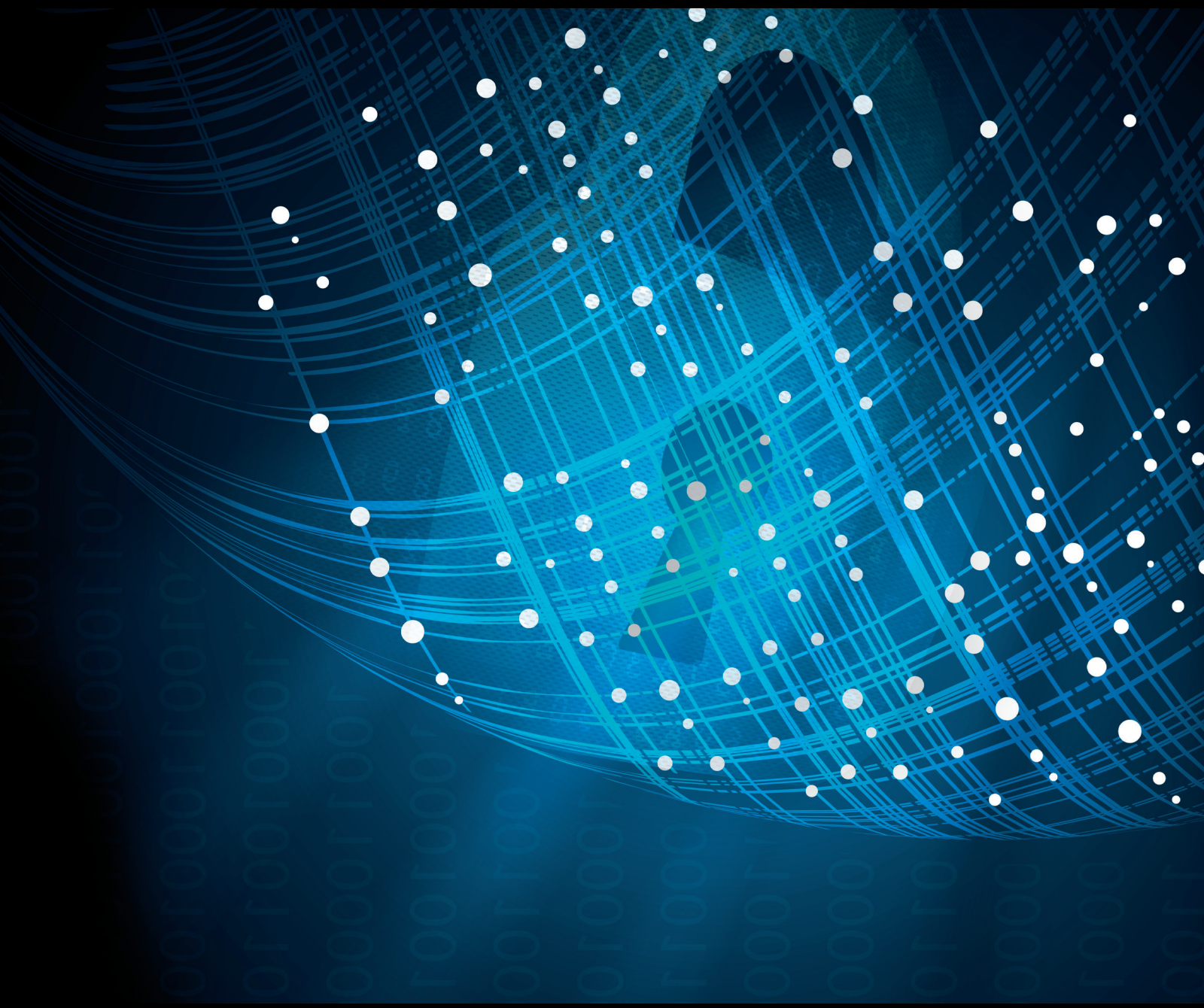
Security and Communication Networks

Privacy and Security of Information Processing in Industrial Big Data and Internet of Things

Guest Editors: Mingwu Zhang, Lein Harn, and Fagen Li



Privacy and Security of Information Processing in Industrial Big Data and Internet of Things

Security and Communication Networks

Privacy and Security of Information Processing in Industrial Big Data and Internet of Things

Guest Editors: Mingwu Zhang, Lein Harn, and Fagen Li

Contents

Hindawi
Security and Communication Networks
Volume 2019, Article ID 5076324, 19 pages
https://doi.org/10.1155/2019/5076324

Research Article

Discovering Vulnerabilities in COTS IoT Devices through Blackbox Fuzzing Web Management Interface

Dong Wang,[1] **Xiaosong Zhang**,[2] **Ting Chen**,[2] **and Jingwei Li**[3]

[1]*University of Electronic Science and Technology of China, ADLab of Venustech, Chengdu, China*
[2]*University of Electronic Science and Technology of China, Chengdu, China*
[3]*University of Electronic Science and Technology of China, State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Chengdu, China*

Correspondence should be addressed to Xiaosong Zhang; johnsonzxs@uestc.edu.cn

Received 16 March 2019; Revised 31 August 2019; Accepted 10 September 2019; Published 4 November 2019

Academic Editor: Prosanta Gope

A novel approach for discovering vulnerability in commercial off-the-shelf (COTS) IoT devices is proposed in this paper, which will revolutionize the area. Unlike previous work, the web management interface in IoT was used to detect vulnerabilities by leveraging fuzzing technology. To validate and evaluate this scheme, a tool named WMIFuzzer was designed and implemented. There were also two challenges: (1) due to the diversity of web interface implementations, there were no existing seed messages for fuzzing this interface and it was inefficient while taking random messages to launch the fuzzing and (2) because of the highly structured seed message, fuzzing with byte-level mutation could conduce to be rejected by the device at an early stage. To address these challenges, a brute-force UI automation was designed to drive the web interface to generate initial seed messages automatically, as well as a weighted message parse tree (WMPT) was proposed to guide the mutation to generate mostly structure-valid messages. The extensive experimental results show that WMIFuzzer could achieve expected result while 10 vulnerabilities including 6 zero-days in 7 COTS IoT devices were discovered.

1. Introduction

With the rapid progress of Internet of things (IoT) technologies, more and more devices have been deployed in our daily lives, such as smart home routers and IP cameras. According to a recent report [1], the number of IoT devices will reach 20+ billion in 2020. These network-enabled devices bring new ability to customers and make their lives easier, while they also attract attacks to compromise them [2–4]. Some studies show that many IoT devices are protection-less, and the security vulnerabilities in them are usually easy to be exploited [5–9]. For example, a security researcher in F-Secure found 18 zero-days in a Foscam IP camera, including insecure default credentials, command injection, stack-based buffer overflow, and so on [10]. Security vulnerabilities in IoT devices can cause a big impact because of the huge quantity of these devices. An example is the Mirai that launched a massive DDoS, which made several leading online services inaccessible including Twitter, PayPal, and Netflix [11]. Therefore, it is crucial to discover the vulnerabilities in IoT devices before deploying them in practice.

A straightforward approach for discovering vulnerabilities is analyzing implementation flaws in the firmware of targeted IoT device, as the firmware provides the low-level control of the device hardware. This is called firmware-based approach that usually contains three steps. Firstly, firmware images are collected from public channels, such as online support service [12]. Secondly, these images are processed by unpacking tools, such as Binwalk, BAT, and FRAK [13–15]. Thirdly, static methods [16–18] or dynamic methods [6, 15, 19, 20] are deployed to detect flaws in these unpacked files. However, firmware-based approaches suffer from known drawbacks. The first one is the availability of firmware images since many vendors do not release them publicly. The second one is the difficulty of unpacking

images, as most vendors prefer to pack them with a compression algorithm to reduce the consumption of device storage. Existing tools can only unpack some images because they work upon known algorithms and file formats. Those images packed with a private file format or encrypted with a private key cannot be unpacked by these tools. For example, a prior work [19] collected 23,035 images, and it only succeeded in unpacking 8,617 images of them via existing tools. The third one is the difficulty of binary analysis due to the diversity of underlying architectures. Different IoT devices usually use different chipsets that have customized features (e.g., instruction sets, memory layouts, and so on). It is still challenging to analyze the unpacked binary files without the knowledge of underlying architecture.

Due to the lightweight and resource constrain, many IoT devices usually provide network interfaces to allow users to interact with them instead of a screen and a keyboard. Web management interface that is designed for device administration is a popular instance of these network interfaces. This interface packs user inputs into a message and sends it to the device. After receiving this message, the device parses the message and does more further procedure according to the message content (e.g., executing a targeted program). If there is an implementation flaw in the message parsing or the further procedure, a vulnerability may be exploited. So, an IoT device that has the web interface can be treated as a blackbox, and feeding this box with malformed messages could trigger potential vulnerabilities of it.

Motivated by this, this paper leverages mutation-based fuzzing technology to perform blackbox testing automatically. Since the web management interface is accessed via network, it is fully independent of the device firmware images and can be used to test COTS IoT devices that do not release their firmware images publicly. Additionally, this blackbox testing does not require the knowledge of underlying architecture about the targeted device. Two challenges detailed in Section 3 are needed to be addressed, including the generation of initial seed messages and the mutation of highly structured message.

To validate and evaluate this blackbox fuzzing, a tool named WMIFuzzer was designed and implemented for discovering vulnerabilities in COTS IoT devices automatically. WMIFuzzer was tested on 7 devices, and it discovered 10 vulnerabilities including 7 zero-days. National Internet Emergency Center in China (CNCERT/CC) is a coordination organization [21] that is responsible for analyzing and reducing cyber threats, vulnerabilities, disseminating cyber threat warning information, and coordinating incident response activities. All vulnerabilities found by WMIFuzzer have been reported to the CNCERT/CC, and all of them have been fixed now. WMIFuzzer was also compared to the state-of-the-art fuzzers, AFL and Sully [22, 23], and the result showed that it is better than both of them in efficiency and effectiveness.

In short, this paper makes the following major contributions:

(i) To the best of our knowledge, WMIFuzzer is the *first* blackbox fuzzer designed for fuzzing the web management interface in COTS IoT devices automatically

(ii) Two challenges about fuzzing this interface were addressed including the generation of initial seed messages and the mutation of highly structured message

(iii) We implemented the blackbox fuzzing in WMIFuzzer and evaluated it on 7 COTS IoT devices: 10 vulnerabilities including 6 zero-days were found

The remainder of this article is structured as follows. In Section 2, the background knowledge of IoT vulnerability discovery is introduced. Then, the design of WMIFuzzer is presented in Section 3. In Section 4, the implementation and evaluation are presented. The limitations are discussed in Section 5. The survey of related work in Section 6 is followed by the conclusion in Section 7.

2. Background

In this section, the background knowledge about discovering vulnerabilities via fuzzing web management interface is introduced.

2.1. Typical IoT Network Architecture. In a typical IoT environment (like the smart home shown in Figure 1), several devices are deployed for different purposes. There are two types of IoT nodes: a *gateway node* and multiple *sensor nodes*. The *sensor nodes* work by (1) collecting external information and pushing them to the remote user and (2) receiving commands from the remote user and executing them. Some *sensor nodes* are coupled with Internet capability, such as the camera connected to the *homeGate* via WiFi. While some cannot access the Internet directly, such as the wristband connected to a mobile phone via Bluetooth and the light connected to a hub via Zigbee. The *mobile phone* and the *Zigbee hub* can make a connection with remote user via Internet, and both of them contain a proxy module that works as a bridge between the Bluetooth (Zigbee) and the Internet. The *gateway node*, which is usually a wireless home router, provides the access point of Internet. The *camera*, *mobile phone*, and *Zigbee hub* are connected to this node. It plays an important role since the insider and outsider of the whole home network are separated by this *gateway node*.

Unlike the traditional information system, there is usually a lot of sensitive data in the IoT environment (e.g., sleeping patterns, health information, human activity, and child privacy). Since these data are shared through the Internet that is an open network, security and privacy are very critical. Due to the resource constrain, some lightweight authentication protocols are proposed [24, 25] for IoT devices. Another recent work proposed a privacy-preserving scheme, called PrivHome, which supports authentication, secure data storage, and query for sensitive data [26]. Since PrivHome is lightweight, it is practical for smart devices which have limited resources.

Most IoT nodes do not provide a screen and a keyboard, and they prefer to provide a network interface to facilitate

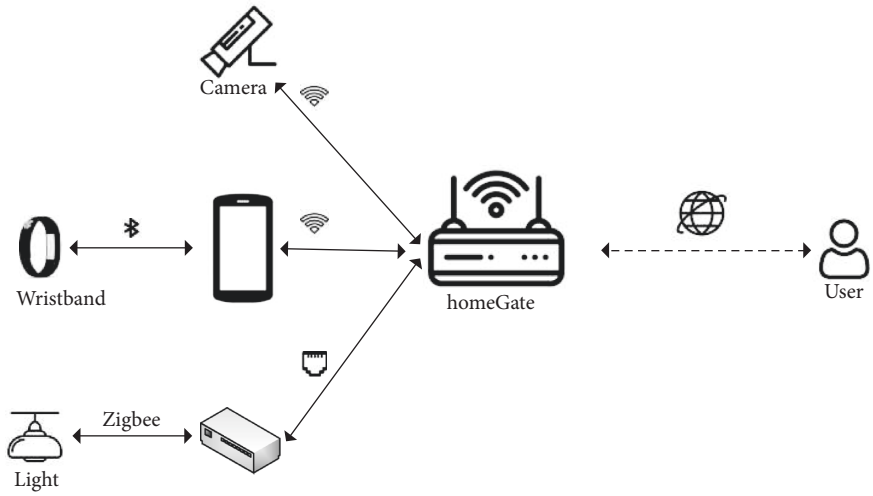


Figure 1: Smarthome network architecture.

the user's operations on device management. For example, the customer uses the browser to login to his camera management interface and update video parameters. If the implementation of this network interface contains security vulnerabilities, an attacker can apply them to launch a crime. A prior study shows that security is an important factor for IoT users [27] because people want the devices to be under their control rather than being fully automated. However, vulnerabilities can make people lose control of these devices.

In this paper, we focus on discovering vulnerabilities in IoT devices that have the capability of Internet since an attacker can do more serious harm by compromising them. For example, vulnerabilities in IP cameras had been used to launch the well-known DDoS. Moreover, even if the camera in Figure 1 does not contain any vulnerability, an attacker can also utilize vulnerabilities in the gateway node to redirect the network flow of the camera into an evil Internet node to extract privacy via machine learning [28, 29].

2.2. Web Management Interface in IoT Devices. As stated previously, an IoT device usually provides a network interface for users to manage itself. Although there are no standards about how to implement this interface, many vendors prefer to utilize web technology because of its flexibility and simplification [6, 30]. However, it is well known that making secure web applications is not a trivial task [31].

The overview of the web management interface is illustrated in Figure 2. There are 3 parts: *FrontEnd*, *Webserver*, and *PageHandler*. The *FrontEnd* is composed of HTML codes, JavaScript codes, CSS codes, and other static resources. The *FrontEnd* is run in a browser, and it has 2 goals. The first one is to be presented as a graphical user interface (GUI) to receive user inputs and clicking, and the second one is packing user inputs into a request message and sending it to the *Webserver*. The *Webserver* and *PageHandler* are both run in the IoT device; the former focuses on message encoding/decoding and the latter focuses on business logic. In detail, the *Webserver* firstly decodes the message to URL and other parameters, then it locates the corresponding *PageHandler* and passes parameters to the handler for further procedure, and finally it encodes the result from the *PageHandler* into a response message and sends it back to the *FrontEnd*. The implementations of *PageHandler* are diverse, such as PHP, Perl, Lua, and CGI-bin [32–34]. If there are flaws in the implementation of the *Webserver* or the *PageHandler*, sending malformed messages to this interface could trigger them.

Additionally, an IoT device is usually designed for a specific purpose, such as the IP camera that focuses on the transmission of captured video data via TCP/IP network. Vendors usually prefer to pay more attention on the improvement of data transmission and video quality, while they just pay little resource on the web management interface including the security. A prior work [5] found a lot of embedded devices are misconfigured in the web management interface by static analysis. Therefore, the web management interface is a good surface to discover vulnerabilities in IoT devices.

2.3. Fuzzing Technology. Fuzzing is an automated random testing technology that was first developed by Takanen et al. [35] to understand the reliability of UNIX tools. The core of fuzzing is to fill targeted tools with random testing data in the goal of triggering flaws. Fuzzing has become widely popular in software testing, and many serious vulnerabilities have been found by this technology [23].

Based on how to generate the random testing data, there are two major categories [36]: *generation-based fuzzing* generates random data by following a data model written manually, while *mutation-based fuzzing* generates random data by mutating an initial data seeds. *Generation-based fuzzing* can generate complicated data fields, such as a field that represents the checksum of a set of bytes. SPIKE, Sulley, and Peach are the state-of-the-art fuzzers built on *generation-based fuzzing*, and all of them are popular in protocol testing [22, 37, 38]. However, it is challenging to deploy generation-based fuzzing when the data model is not available, since writing data model is laborious, time-consuming, and error-prone. While *mutation-based fuzzing*

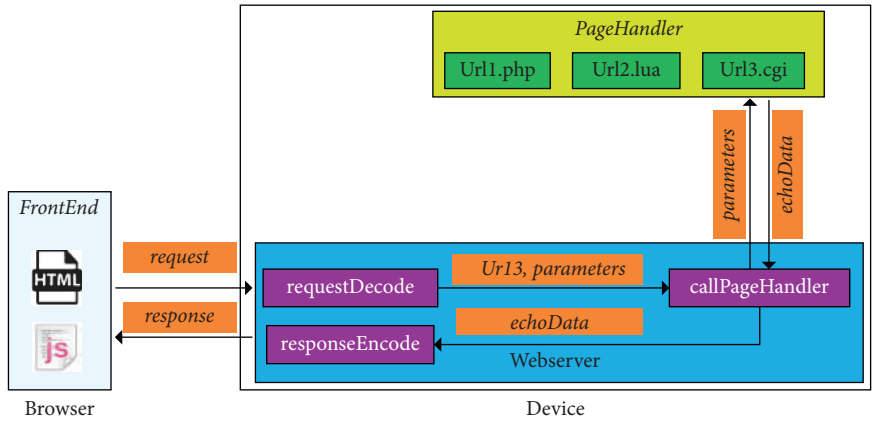


Figure 2: Web management interface.

performs a small mutation on a set of well-formed seeds and then feeds the targeted program with the mutated seeds to monitor unexpected behaviors. For testing COTS devices that do not have documents about its web management interface, it is unpractical to write the data model for every targeted device. In contrast, collecting a set of seeds for every device to start mutation-based fuzzing is much easier.

Based on the complexity, there are three major categories of mutation-based fuzzing [39]: *blackbox fuzzing* just feeds the target with mutated data and monitors the program status in the goal of detecting abnormal behaviors; *greybox fuzzing* employs some feedback information (e.g., code coverage) to improve efficiency; and *whitebox fuzzing* usually performs heavy program analysis (e.g., symbolic execution) to retrieve more internal status to guide the fuzzing. *Whitebox fuzzing* has a well-known drawback of scalability about targeted program [40], so it is challenging for testing real-world programs. *Greybox fuzzing* usually needs to make modification on the original program (source code or binary code) [41], and it cannot work for those programs that cannot be modified. For the web management interface of COTS IoT devices, the *Webserver* and *PageHandler* are embedded inside the device. This constraint makes *whitebox fuzzing* and *greybox fuzzing* to not work. In contrast, *blackbox fuzzing* is independent of program size and it does not require program modification. So, it is practical to discover vulnerabilities in the web management interface by blackbox fuzzing.

3. WMIFuzzer

In this section, we firstly introduce the challenges in WMIFuzzer via an example in Section 3.1, secondly we present the system architecture in Section 3.2, thirdly we present the detailed design about seed generation in Section 3.3 followed by the WMPT for representation of the highly structured message in Section 3.4, and finally we present our fuzzing upon the WMPT in Section 3.5.

3.1. Challenges in WMIFuzzer

3.1.1. An Example. To better understand the problem, we use an enterprise gateway as the example. This device has a web management interface for administration, and its firmware is available publicly. However, the firmware may be encrypted with a private key or be compressed with a private format since it cannot be unpacked via existing tools. So, it is unable to discover vulnerabilities in this device via firmware analysis.

As stated previously, the *FrontEnd* of web management interface is run as a set of GUIs in a browser. Figure 3 illustrates one GUI designed for configuring the *TCP/UDP session limitations* and the corresponding message generated by this GUI. As can be seen from it, this message is highly structured. It contains two parts: *header* and *contents*, and both of them are following the HTTP protocol specification [42]. In the *header* part, *conn_limit.cgi* represents the targeted *PageHandler*. If the *Webserver* makes an incorrect assumption about the length of *PageHandler*, a memory overflow flaw may be triggered by mutating it with a long string. In the *contents* part, *COMN_LIMIT_NUM = 1000* represents the parameter about *TCP session limitation*. If *PageHandler* assumes that it is a number, an unknown bug may be triggered by mutating it with *COMN_LIMIT_NUM = ABCD*. To discover vulnerabilities in this device automatically, one possible approach is mutating every message generated by all GUIs and sending them to the targeted device. However, two challenges are required to be solved in order to launch this mutation-based fuzzing.

3.1.2. Challenge 1: Generation of Initial Message Seeds. Mutation-based fuzzing works upon some well-formed seeds [23], or it will generate a lot of invalid test cases to make the fuzzing inefficient. For fuzzing traditional file format processing software (e.g., BMPViewers), a BMP file can be used to test different BMPViewers since all BMPViewers follow the same format specification. In contrast, an existing web management interface message cannot be used for testing device devices as there is no unique specification for the implementation this interface. For example, the field named *CONN_LIMIT_NUM* in Figure 3 is specific for this example device, and it will be invalid for another device. So, an automatic approach is needed for generating the initial valid message seeds when testing different COTS IoT devices.

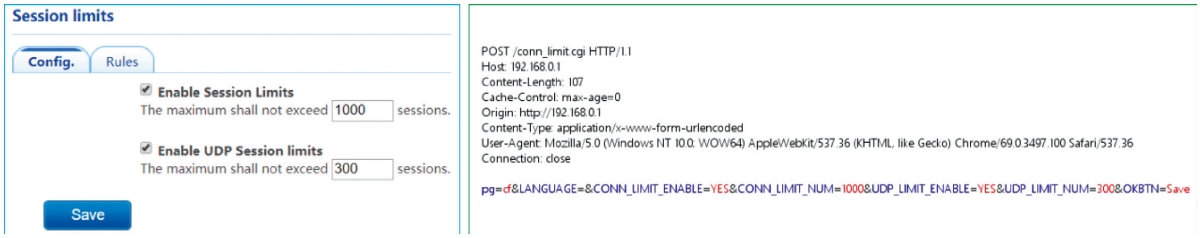


Figure 3: Example of web management interface GUI and message.

3.1.3. Challenge 2: Mutation of Highly Structured Message. Traditional mutation-based fuzzing mainly focuses on unstructured data, most fields of which are at fixed location and have a fixed length. For example, the field about the BMP file length is located in offset 2 with 4 bytes, and its value can be from 0 to max integer number. No matter how the byte-level random mutation changes this field, all BMPViewers still treat this 4 bytes field as an integer number and parse it. In contrast, most fields in the highly structured web management interface messages are not at fixed location and their length is also not fixed. Byte-level mutation on these messages usually corrupts their structure to cause mutated messages being rejected by the device at an early stage. For example, the *POST* field in Figure 3 is the HTTP method, and mutating its byte *S* to the blank ASCII char will cause the remaining *ST* to be a new field named URL followed by the old URL. The *Webserver* will discard this mutated message to stop more deep procedure since it contains two URL fields. Therefore, an approach is required for handling the mutation of highly structured messages.

3.1.4. Solutions. Fortunately, this paper obtained the following approaches to address the above challenges:

(i) *Generating Message Seeds via UI Automation.* UI automation [43] is an automatic method to emulate user operation such as inputting data to UI elements and clicking buttons to submit the inputting. Observed on this, UI automation can be applied to input data and click a button to drive the *FrontEnd* to generate a web management interface message. Since the network of a browser can be intercepted, the generated message can be captured as a seed. By trying all GUIs via UI automation, sufficient messages can be generated as the fuzzing seeds.

(ii) *Mutating a Message Seed Based on Its WMPT.* Abstract grammar tree (AST) [44] is a tree representation of the abstract syntactic structure of source code written in a programming language. Each node of the tree denotes a construct occurring in the source code. The syntax is "abstract" in the sense that it does not represent every detail appearing in the real syntax, but rather just the structural, content-related details. Inspired by the AST and HTTP protocol, dissecting a message to a tree and modifying the content of a node can keep the structural information and perform mutation simultaneously. In this way, the weighted message parse tree (WMPT) is proposed to perform the mutation to generate mostly structure-valid messages. This will bring the blackbox fuzzing the ability to detect deep vulnerabilities in IoT devices, such as the *PageHandler*.

3.2. System Architecture of WMIFuzzer. The architecture of our WMIFuzzer is illustrated in Figure 4. At a high level, there are 3 parts: seed generation, WMPT parsing, and fuzzing upon WMPT. Their goals are presented as follows:

(i) *Seed Generation*. This is the first part that utilizes UI automation to generate seed messages as much as it can. In detail, it firstly tries to explore every GUI via a browser and then triggers GUI events to drive the browser generating original messages. A HTTP proxy is deployed to intercept these messages sent by the browser and save them as the initial seeds. All of these steps are fully automatic.

(ii) *WMPT Parsing*. This part is converting an original message to its WMPT detailed in Section 3.4. The goal of the WMPT is to store the message structure into a tree and store the content of message fields into the tree nodes. In this way, mutation on the nodes of a WMPT will mostly generate structure-valid messages.

(iii) *Fuzzing upon WMPT*. The last part is performing fuzzing schedule on the WMPT generated previously. It performs random mutation to generate structure-valid messages and sends them to the device under testing. To infer whether a vulnerability is triggered, several monitoring strategies are deployed to monitor the device running status.

3.3. Seed Generation via UI Automation. As stated previously, WMIFuzzer is built on mutation-based fuzzing that needs a set of valid seeds to start the fuzzing. This section describes how to generate valid messages as the initial seeds by UI automation.

Definition 1. Traditional UI automation widely used by software developers is a 5-tuple $T_{ui} = (I, S, O, E, M)$, where

(i) I is the set of GUI elements used to receive user inputs in which $\alpha \in I$ is called input control (IC)

(ii) S is the set of GUI elements used to receive user clicks in which $\beta \in S$ is called submission control (SC)

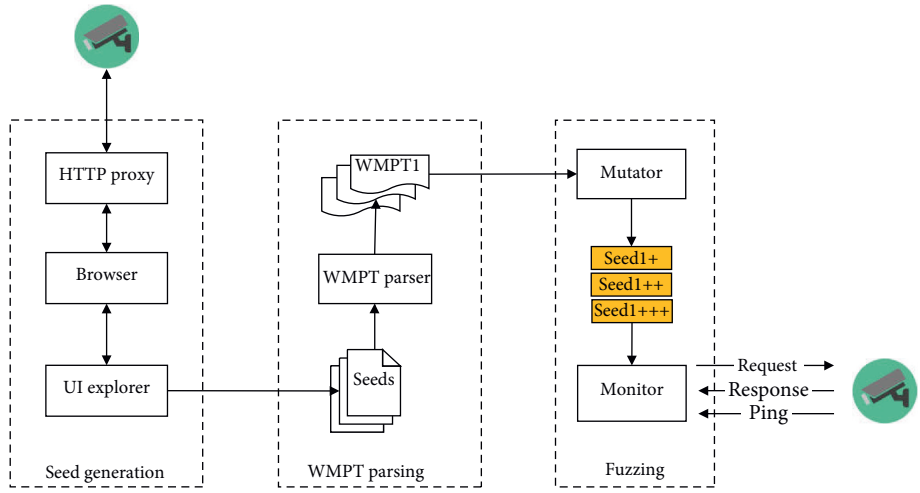


Figure 4: Overview of WMIFuzzer.

(iii) O is the set of GUI elements that cannot receive user inputs and click in which $\delta \in O$ is called other control (OC)

(iv) E is the set of all GUI elements in which $E = I \cup S \cup O$, $I \cap S = \phi$, $I \cap O = \phi$, and $S \cap O = \phi$

(v) M is the set of all mappings: $\{M_j \mid I_j \longrightarrow \gamma_j\}$ in which $I_j \subseteq I$ and $\gamma_j \in S$

The procedure of UI automation is enumerating every $\gamma_j \in S$, inputting data to I_j, and clicking γ_j. Figure 5 is one example web management interface GUI that has two SCs named *Save* and *Check again*. This GUI has two mappings: (1) the *Save* is mapped to fours ICs named *Enable, Disable, 02,* and *00* and (2) the *Check again* is mapped to none of these ICs. With these mappings, UI automation firstly inputs four values to these four ICs and then clicks the *Save* or directly clicks the *Check again* without inputting.

In practice, the mapping M in Definition 1 is generated by software testing engineers by hand. However, an automatic approach for building the M is required since WMIFuzzer is designed to fuzz different IoT devices automatically. A previous work [45] that utilizes UI automation to detect IoT account vulnerabilities builds this mapping by the observation that many IoT control apps share the same style for the account management GUI, and most of these apps use the default ICs and SCs. In contrast, the GUI of web management interface in IoT devices is diverse, as different types of devices are usually designed for different purposes. Even for the same type of devices, most vendors prefer a new GUI style to make themselves different from other vendors. So, WMIFuzzer cannot depend on the GUI style, and it has to analyze the codes of *FrontEnd* to identify I, S, and M.

In the development domain of *FrontEnd*, the browser renders the GUI based on three parts: HTML codes, JavaScript codes, and CSS codes. HTML is the standard markup language for creating web pages, and HTML elements are the building blocks of a web page. The JavaScript codes process the business logic (e.g., submitting user inputs), while the CSS codes describe the style of E. This means that every $\kappa \in E$ is declared as an HTML element in the HTML code. According to the HTML specification [46], different types of HTML elements have different purposes: *input* and *textarea* elements are defined as ICs and *input-with-submit* and *button* elements are defined as SCs. Therefore, WMIFuzzer should identify I and S from all elements in the HTML code. Unfortunately, it usually fails because of the flexibility of HTML elements. In Pseudocode 1 we present the core HTML codes of the GUI presented in Figure 5.

From the above codes, we can get the following information:

(i) *Check again* can be identified as a SC since it is a button element in code line 3.

(ii) *Enable* and *Disable* cannot be identified as ICs since they are *img* elements in code lines 5-6, but the *FrontEnd* utilizes two *img* elements with boolean-style value attribute to work as a checkbox-style IC. So, these two *img* elements are not ICs, but they have the same functionality of ICs.

(iii) *02* and *00* cannot be identified as ICs because they have the read-only attribute in code lines 9–12, but their parent element has a click event that can dynamically modify the read-only attribute.

(iv) Lines 15-16 contain two useless ICs that have hidden attributes to make themselves invisible in the GUI.

(v) *Save* also cannot be identified as a SC since it is a *div* element in code line 17, but this element has been binded with a click event handler in code line 19. The event handler collects data from two read-only elements and saves data to a hidden element hosted in a form in code lines 20-21. At last, the event handler triggers the form submitting a request in code line 22. So, this *div* element has the same functionality of a SC.

In short, every HTML element can be an IC or a SC because it can use value attribute and event handler to receive user inputs and button clicks. This is the flexibility of

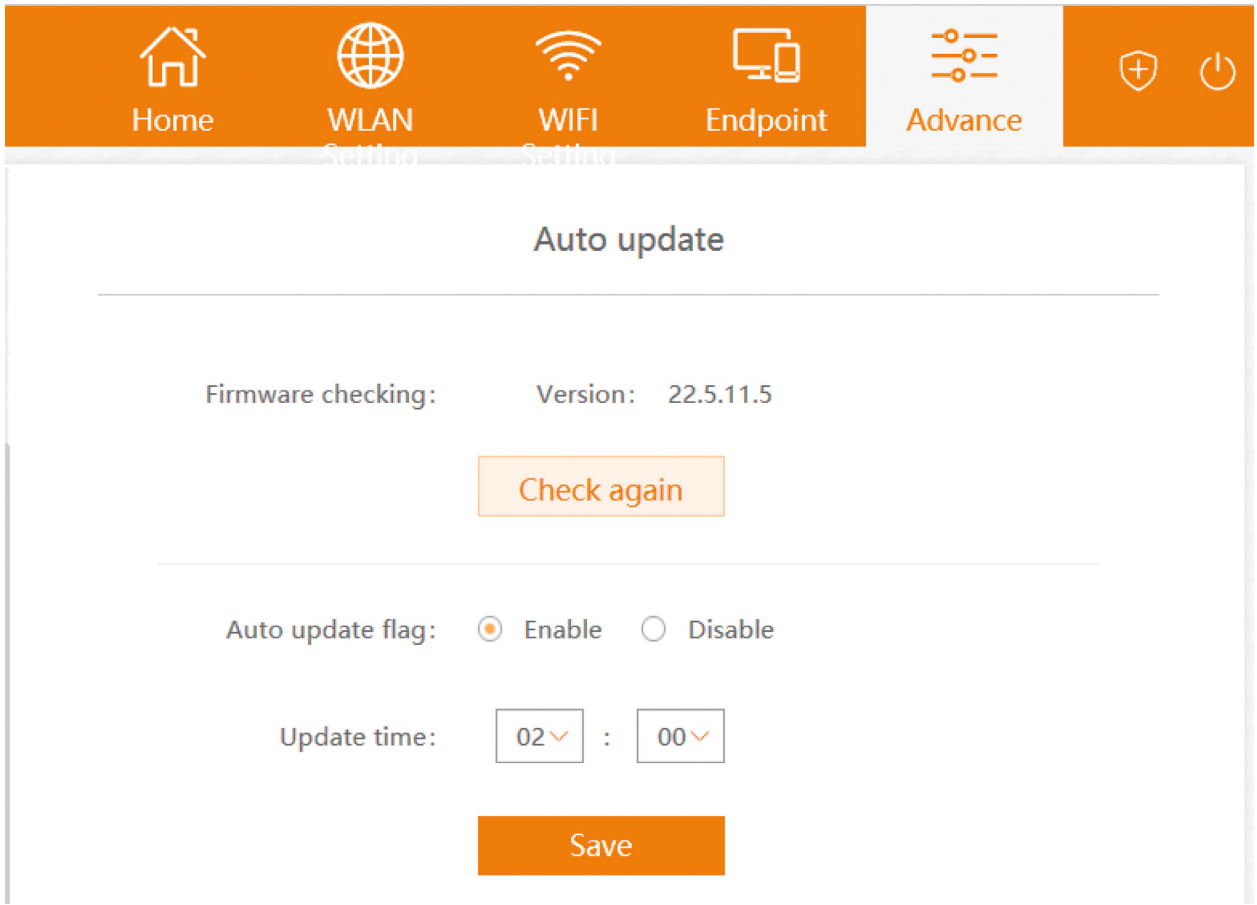


Figure 5: GUI of the AutoUpate.

```
(1) <div><span>Auto update</span>
(2) <div><span>Firmware checking: Version: 22.5.11.5</span>
(3) <button id = "recheck_btn">Check again</button></div>
(4) <div><span>Auto update flag: </span>
(5) <img src = "/radio_on.png" value = "1"><span>Enable</span>
(6) <img src = "/radio_of f.png" value = "0"><span>Disable</span>
(7) <span>Update time: </span>
(8) <div onclick = "dropdown_click (this)" type = "button">
(9) <input id = "uptimehour" Type = "text" read only = "read only"
(10) value = "02"></div>
(11) <div onclick = "dropdown_click (this)" type = "button">
(12) <input id = "uptimemin" Type = "text" read only = "read only"
(13) value = "00"></div>
(14) <form action = "/autoupgrade/save" method = "post">
(15) <input type = "hidden" value = "1">
(16) <input name = "autoUpTime" type = "hidden" value = "·"></form>
(17) <div id = "AutoUpSave" type = "button">Save</div>
(18) </div></div>
(19) $("#AutoUpSave").click (function (·){
(20)     $ ("#autoUpTime").val ($ ("#uptimehour").val (·)+":" +
(21)     $ ("#uptimemin").val (·));
(22) setTime out (function (·) {document.save.submit (·);}, 1000);
(23) });
```

Pseudocode 1: AutoUpdate.html.

web technology, and it makes identifying the ICs and SCs difficult. Program analysis (e.g., symbolic execution and taint tracking) can be utilized to explore whether a HTML element can collect data or submit data to help identify the ICs, SCs, and their mappings, but their known limitations make them inefficient for the web interface.

As it is a challenge to identify ICs, SCs, and their mappings, a brute-force UI automation that makes a trade-off between the accuracy and the practice is proposed.

Definition 2. Brute-force UI automation is a 2-tuple $T_{\text{ui}} = (E, M)$, where

(i) E is the set of all GUI elements in which $\alpha \in E$ is an IC and a SC at the same time

(ii) M is the set of all mappings: $\{M_j \mid E_j \longrightarrow \gamma_j\}$ in which $E_j = E - \gamma_j$ and $\gamma_j \in E$

This means (1) every element is an SC, (2) every element is an IC, (3) and every SC is mapped to all ICs. So, M is very simple, and its size is a constant that matches the element count of E. It seems that this brute-force UI automation works well at most times, since the event handler of most SC ignores those ICs that are not really mapped to this SC. In this way, WMIFuzzer can use UI automation to drive the *FrontEnd* to generate messages automatically by the following steps:

(i) *Initialization*. In this step, the UI automation firstly starts a browser, which is run under control, to open a welcome page that guides the user to input the IP address of the targeted IoT device. Then, the browser opens the login page of the web management interface to guide the user inputting his credentials to login to the interface. This is the only step that requires a little of manual efforts because this interface usually denies accesses without successful login. On the contrary, this is also the standard step for managing an IoT device via web management interface. The whole step does not require any knowledge about software testing and vulnerability discovering.

(ii) *Crawling*. The web management interface is defined and implemented by the device vendor, so there are usually not any documents about how many web GUIs are there in a device. For a customer, he/she accesses the root GUI via the IP address of the device and then switches to other GUIs by exploring the root GUI. As all GUIs are accessed by URLs, WMIFuzzer performs crawling from the root GUI to get all GUI-related URLs. A GUI may (1) not send a network message as it is a static GUI, (2) or send one network message as it is a simple GUI, (3) or send multiple network messages as it is a complex GUI with multiple business logics. So, heuristic strategies are deployed to exclude mostly useless URLs: (1) URLs that do not start with the IP address of the IoT device since the messages with these URLs will not be sent to the device and (2) URLs that have a filename extension about known names (e.g., png, ico, bmp, jpg, CSS, xml, and js) since they are not GUIs but pure static resources.

(iii) *Inputting and Clicking*. For every URL collected in the crawling step, firstly its HTML codes are analyzed to identify all elements, then an initial value is brute-force assigned to every identified element, and finally an element is chosen to be clicked by injecting a brute-force click. WMIFuzzer chooses the click element by the elements' sequential orders in the HTML codes, and the values of all elements are reset every time before WMIFuzzer chooses the click element. Moreover, all HTML elements of a URL are just identified once and the result is saved. This feature enables directly locating all elements and inputting values to them when this URL is reloaded.

(iv) *Capturing Messages*. Previous clicking may trigger the browser generating a network message sent to the device, and the message can be built in the JavaScript codes (e.g., posting data via Ajax) or in the standard form submission. To capture these messages, API hooking [47] can be utilized to intercept the network-related APIs in the browser, but API hooking is not a robust approach, so another approach named *browser network proxy* that is supported by most browsers natively is utilized in WMIFuzzer. By installing a proxy and configuring the browser with this proxy, all messages will be transparently sent to the proxy instead of the real device.

In a word, WMIFuzzer overcomes the challenge of seed generation by utilizing the brute-force UI automation.

3.4. Weighted Message Parse Tree. Traditional mutation-based fuzzing works by randomly selecting a set of bytes in a seed, modifying them and feeding the target with this mutated seed. However, it cannot work well for the web management interface messages that are highly structured, as pure byte-level random mutation mostly generates structure-invalid messages to make the fuzzing inefficient. Inspired by the abstract syntax tree [44, 48], WMIFuzzer applies the WMPT to represent the seed message and perform random mutation on the WMPT.

A WMPT is an ordered rooted tree with typed nodes where each node has a weight scope, and this tree has two types of nodes: leaf nodes and internal nodes. A leaf node contains an atomic part of message content while an internal node concatenates its child nodes. The weight of a node means how much time should be assigned to the mutation of this node. Figure 6(a) is an example message about web management interface and Figure 6(b) is the corresponding WMPT. All green nodes are the leaf nodes that store the atomic content of a seed message, and others are internal nodes that store the structure of their child nodes. Obviously, the mutation of a leaf node will not change the structure of the WMPT at most times. Moreover, inserting or

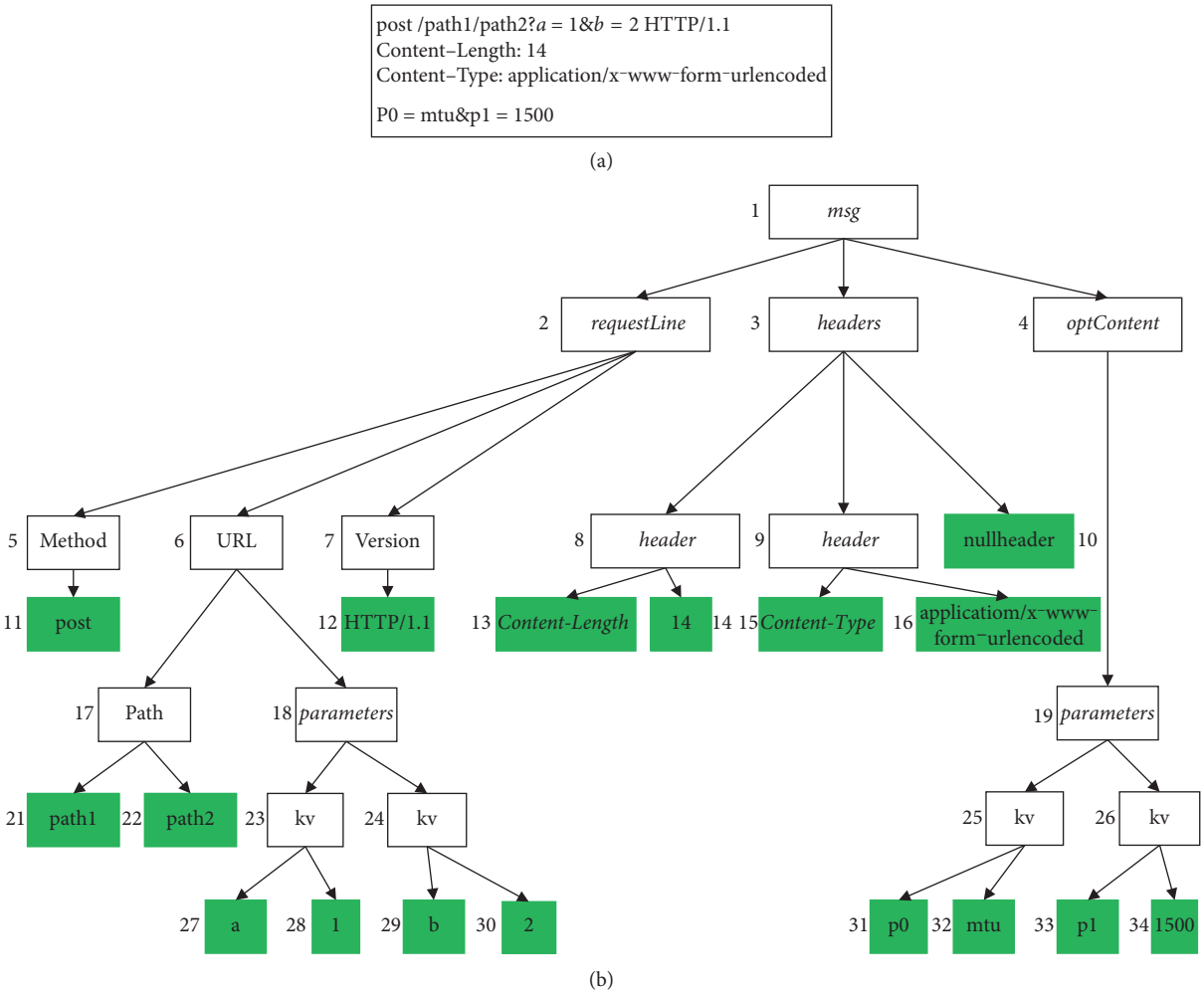


Figure 6: Weighted message parse tree. (a) A sample message. (b) The WMPT of the sample message.

deleting some internal nodes can generate mutated messages that have a big difference from the original seeds. For a mutated tree, collecting the content of its nodes and contacting them will output a message.

Based on the mutation of leaf nodes and internal nodes in WMPT, WMIFuzzer can generate both structure-valid and structure-invalid messages on demand. So, it is capable to detect shallow and deep vulnerabilities. There are three operations on the WMPT: *deserialization* is converting a seed message to be a WPMT and *mutation* is modifying the WMPT while *serialization* is converting the mutated WMPT to a new message.

3.4.1. Deserialization Operation. As the web management interface messages are built on HTTP protocol, WMIFuzzer follows the protocol specification to dissect a seed to its WMPT. Algorithm 1 details the procedure of seed deserialization. For every seed message $\alpha \in M$, WMIFuzzer needs to perform deserialization for converting it to a WMPT $t \in T$. According to the HTTP protocol specification [46], the web interface message is an ACSII string composed of 3 parts: *requestLine*, *headers*, *and optContent*. The *requestLine* is the first line of the message, the *headers* are the following lines that end with a special empty line, and the *optContent* is the whole remaining data determined by some special line in the *headers*. So, Algorithm 1 firstly splits a message into these three parts (nodes 2, 3, and 4 in Figure 6) since this is the mandatory constraint about HTTP message.

The *requestLine* contains three parts: a request *method* that is a predefined ASCII string (e.g., GET, POST, and so on), a targeted URL that is composed of *pagePath* and optional *parameters*, and a predefined protocol *version* (nodes 5, 6, and 7 in Figure 6). The two parts of a target URL are concatenated via the symbol "?" (nodes 17 and 18 in Figure 6). The *pagePath* is separated by the symbol "/" (nodes 21 and 22 in Figure 6), while the *parameters* are encoded as k-v blocks concatenated via "&" (nodes 23 and 24 in Figure 6) and k-v is concatenated via the symbol "=" (nodes 27 and 28 in Figure 6). In this way, Algorithm 1 generates the *requestNode* and inserts it to the WMPT *t*. Since the targeted URL is the most customized part, a double weight scope was assigned on its child nodes to perform more mutation.

The *hea de rs* are a set of header lines, that is, a k-v style string, and the separator is a colon. There is no limitation of

Input: the set of seed messages, M;
Output: the set of generated WMPT, T;
(1) initial $T = \varnothing$
(2) **for** each $\alpha \in M$ **do**
(3) initialize an empty WMPT $t = \varnothing$
(4) split α to 3 parts {*requestLine*, *headers*, *optContent*}
(5) generate *requestNode* from *requestLine* and insert it to t
(6) initialize an empty *headersNode* = $\varnothing$
(7) **for** echo $\beta \in$ *headers* **do**
(8) generate header from β and insert it to *headersNode*
(9) **end for**
(10) insert *headersNode* to t
(11) if *optContent* is not empty **then**
(12) generate *contentNode* from *optContent* and insert it to t
(13) **end if**
(14) **end for**
(15) **return** T

Algorithm 1: Seed dissection algorithm.

the quantity of header lines, and the HTTP protocol applies an empty line as the last header line. There are some standard header lines with known keys generated by the browser, such as *Content-Length, Content-Type, Cookie, User-Agent*, while other customer header lines are generated by the *FrontEnd*. To parse each line, we also follow the k-v style to decode the line. Because the data format of the value content in a k-v header line is not known, heuristic strategies are applied to parse the content into basic elements by scanning special chars including blank, comma, semicolon, and "&". In this way, Algorithm 1 generates the *headersNode* and inserts it to the WMPT t.

The *optContent* is an optional part, and there is no limitation of its content. The content length is declared in a well-known header named *Content-Length* in *headers*, and it is strongly recommended to declare the content type in another well-known header named *Content-Type*. In the web development domain, there are 3 popular types: application/x-www-form-urlencoded, multipart/form-data, and application/json. For these three types, WMIFuzzer deserializes the content following their specifications, while for other types, the content is deserialized as a bytes stream. With this, Algorithm 1 generates the *contentNode* and inserts it to the WMPT t. Finally, a message is dissected to its WMPT.

3.4.2. Mutation Operation. As stated previously, the mutation is performed on the WMPT nodes to generate mostly structure-valid messages. For a leaf node, WMIFuzzer mutates its content with random data directly. Since the HTTP message is built on text string that ends with a NUL character (NUL has all bits zero), NUL is imported into a mutated message before the tail will make the message to be truncated. For example, node 7 in Figure 6 is the protocol *version*, and the *headers* and *optContent* will be truncated by the *Webserver* in IoT devices when a char of this node content is mutated to NUL. So, NUL is filtered when performing mutation. In detail, when mutation will change a byte to NUL, WMIFuzzer dynamically generates a random number, that is, 1100 and checks whether it is bigger than a threshold (now is 97). If the result is TRUE, WMIFuzzer accepts this mutation, or it just ignores this mutation. For an internal node, WMIFuzzer deletes the node or deletes a child node randomly to the mutation on a leaf node, this mutation will bring a big difference.

3.4.3. Serialization Operation. Serialization is converting a WPMT to its corresponding message, and it is very straightforward: traversing the WPMT depth-first and contacting the content of every leaf node. The contacting symbol is determined by the parent node of every node. Especially, the serialization will keep the parent node symbol when a leaf node is mutated to an empty value. For example, node 30 in Figure 6 is a parameter value in URL, and the serialization of its parent (node 24) will be "b=": In contrast, serialization will not keep the parent node symbol when a leaf node is mutated to be deleted. For the previous example, the serialization of node 24 will be *b*.

3.5. Fuzzing upon WMPTs. Based on the generated WMPTs, the *Mutator* in Figure 4 can be scheduled to generate structure-valid mutated messages. Then, these mutated messages are sent to IoT devices in the goal of triggering vulnerabilities detected by the *Monitor*.

3.5.1. Fuzzing Scheduling. There are multiple nodes in a WMPT, and it is unknown that which node with mutated content can trigger a vulnerability. So, scheduling as shown in Algorithm 2 is proposed to perform the fuzzing. There are two phases of fuzzing: determined phase and random phase.

The determined phase is performing limited mutation on a single leaf node to trigger vulnerabilities as fast as it can. Every seed message $\alpha \in M$ will be processed once in this phase. For a leaf node $\beta \in \alpha$, *sequentia_mutating* tries following modifications including

```
Input: the set of generated WMPT, M;
(1)  for each α ∈ M do
(2)    for each node β ∈ α do
(3)      if β is a leaf node then
(4)        tm = sequentia_mutating(β)
(5)        result = sending_monitoring(tm)
(6)        if interesting(result) then
(7)          alert(tm)
(8)        end if
(9)      end if
(10)   end for
(11) end for
(12) loop
(13)   for each α ∈ M do
(14)     indexList = randomIndexList(0, len(α))
(15)     for each λ ∈ indexList do
(16)       tm = random_mutating(λ)
(17)     end for
(18)     result = sending_monitoring(tm)
(19)     if interesting(result) then
(20)       alert(tm)
(21)     end if
(22)   end for
(23) end loop
```

Algorithm 2: WMIFuzzer fuzz scheduling algorithm.

(i) Extending content in the goal of triggering vulnerabilities about buffer overflow. WMIFuzzer just appends a number of printable ASCII chars to the content for simplification. Given the limitation of memory resource in IoT devices, the length of extended content is configured to classic values: 2^i, where $0 \le i \le 12$.

(ii) Emptying the content in the goal to trigger vulnerabilities about assumption of nonempty content. For example, the serialization of the URL parameters will be $a =$ and $b = 2$ when node 28 in Figure 6 is mutated to empty content. As the content owned by the key a is empty now, an implementation flaw may be triggered if the *Webserver* or the *PageHandler* makes an incorrect assumption.

(iii) Replacing the numerical-style content with typical integer number in the goal to trigger vulnerabilities about integer overflow. Because the HTTP message is text based and there is no standard approach to identify whether the node content is a string or a number, WMIFuzzer just applies regular express to match whether the content looks like an integer number. If the matching result is FALSE, it will be treated as a string, or the content will be replaced with classic boundary integer numbers: 2^i, $2^i - 1$, and $2^i + 1$, where $0 \le i \le 32$.

(iv) Replacing the string-style content with special strings in the goal to trigger various vulnerabilities. FuzzDB [49] is an open-source project that collects a lot of popular strings used in multiple security tools, and WMIFuzzer also uses its string database to replace string-style content.

(v) Changing the content type in the goal to trigger vulnerabilities about assumptions on the data type. For the node that matches an integer number, it will be replaced with an ASCII string, or the node content is treated as a string and will replaced with an integer number. For example, node 28 in Figure 6 is an integer string, and the *Webserver* will convert it to an integer number. However, the conversion may fail when the integer string is mutated to a nonnumerical string, and a crash may occur when the *Webserver* does not check the result of conversion.

In contrast, the random phase is an endless loop that performs random mutation on all seeds repeatedly. When fuzzing a WMPT, only a subset of its nodes will be mutated at a time. This is because the generated message will be useless and rejected at an early stage at most times if most nodes are mutated. The *randomIndexList* in Algorithm 2 is firstly used to generate a list of index values that identify which nodes should be mutated in this iteration. And the *random_mutating* is used to perform mutation as shown in following:

(i) For a leaf node, both the modifications in the determined phase and random byte-level are fully supported. In contrast to the determined phase, the major difference is that every mutated message in this phase may contain modification on multiple nodes.

(ii) For an internal node, deleting itself and deleting a child node are both supported. As there are multiple mutations on different nodes via *indexList*, deleting a node from the tree will make the *indexList* invalid and the remaining mutations may trigger a crash. Inspired by the *copy-on-write* [50] that is popular in modern operation system (e.g., Windows and Linux), WMIFuzzer does not really remove the node from the tree, and it just marks the node 'deleted' to notify the serialization of WMPT ignoring this node.

The messages generated in these two phases will be sent to the IoT devices under testing. If an interesting event in the following *Remote monitoring* is detected, an alert will be generated with the current mutated message to help further manual inspection.

3.5.2. Remote monitoring. Monitoring running status is very important because it is the only way to infer whether a vulnerability is triggered in a COTS device. However, it is challenging to monitor the running status since these devices usually do not support local monitoring to make a lot of information unavailable (e.g., running processes, threads, CPU overhead, and memory statistics). So, the main surface for device monitoring is the network communication. If a vulnerability causes the device to crash, the device network will be unavailable and this can be detected in the outside of the device. But not every vulnerability will cause the system crash, such as the *Command Injection* and *Interface Leak*. If *Command Injection* executes a network-related system command in the device, the network status of device will also

be infected and this can be monitored outside. So, WMIFuzzer uses two commands, *reboot* and *ping* to infect the device network; the former can make the device to go offline and the latter can send *ICMP* message. The root cause of *Interface Leak* is incorrect authentication, and WMIFuzzer uses a simple strategy to monitor this type of vulnerabilities. If a mutated message contains modification in the authentication data or in the URL but the response status is successful, a leak vulnerability is triggered. Given the device has two phases to consume the mutated messages, *Webserver* and *PageHandler*, we analyze the potential network status when a flaw is triggered in each phase:

(i) *Webserver Phase*. If a message just crashes the *Webserver*, there is no response message after the mutated message is sent to the device, and the *Webserver* may become unavailable if it is running in a single-process model while the device low-layer network is available. Moreover, if the crash also infects the system at the same time, the device may reboot due to the following reasons: 1) no response , 2) Webserver unavailable, and 3) low-layer network unavailable. Otherwise, the *Webserver* will reply a response message, and both the low-layer network and *Webserver* are reachable.

(ii) *PageHandler Phase*. If a message crashes the *PageHandler*, the fuzzer will not receive the response message while the *Webserver* is available. Moreover, if the crash also infects the system, the network status is the same as the crash in the *Webserver* and they cannot be distinguished from each other.

In a word, WMIFuzzer monitors the status of the low-layer network, *Webserver*, and *PageHandler* to detect multiple types of vulnerability including crash, *Interface Leak*, and *Command Injection*.

4. Implementation and Evaluation

We present the prototype implementation of WMIFuzzer in Section 4.1 and the evaluation in Section 4.2 and then briefly study some vulnerabilities found by WMIFuzzer in Section 4.3.

4.1. Implementation of WMIFuzzer. WMIFuzzer was implemented with around 3,000 Python lines of code and 2,000 C lines of code in total. Also, several open-source projects (e.g., Chrome, Selenium, mitmproxy, and fuzzdb [49, 51, 52]) are integrated into this fuzzer to avoid reinventing the wheel.

In the seed generation phase, the brute-force UI automation was built based on Chrome and its Selenium driver. Python code was written to use the Selenium driver to control the Chrome behavior, such as opening a URL, inputting data, and clicking a button. The mitmproxy project, a HTTP proxy written in Python code, was extended to intercept the web management interface messages.

In the WMPT parsing phase, C code was written to implement the deserialization, mutation, and serialization. HTTP protocol specification was used to parse the URL, headers, and content. To reduce redundant parsing of the content type of a node, the data type is calculated once when converting a message to its WMPT and saved into the node. Inspired by the copy-on-write strategy, the mark-deleted flag was applied on the WMPT nodes to reduce unnecessary memory operations. The C code was compiled to a dynamic library that can be loaded by Python code.

In the Fuzzing phase, Python code was written to schedule the fuzzing. After a mutated message has been sent to the device, the reachability of *Webserver*, *PageHander*, and the low-layer network is checked to detect crashes and command injection in the targeted device. In addition, the response message was analyzed to detect interface leak.

4.2. Evaluation of WMIFuzzer. In order to evaluate the effectiveness and efficiency of WMIFuzzer, we tested it on 7 IoT devices and compared it with two state-of-the-art fuzzers: American Fuzzy Lop (AFL) [23] and Sulley [22].

4.2.1. Testing Devices. We bought 7 popular IoT devices, and all of them contain the web management interface. After manually checking their firmware images, 4 of them have been released publicly while others were not. For the public firmware images, popular tools were deployed to unpack them: Binwalk [14] and BAT [15]. And just one of these images can be unpacked, but it fails on being deployed in an emulator presented in a previous work [19]. The detailed specifications of these devices are described in Table 1, where FirmwareAva means firmware is released publicly, FirmwareDec means firmware can be unpacked, and FirmwareRun means firmware can be run in an emulator environment.

4.2.2. Testing Environment. WMIFuzzer and the other two fuzzers are run in a separate virtual machine that hosts an Ubuntu 16.04 with Intel Core i7 quad-core 3.6 GHz CPU with 4G RAM. The original branch of AFL cannot work on the web management interface because it works by testing targeted program locally. So, its code was patched to create two versions: AFL^1 and AFL^2. AFL^1 replaced its code about target program execution with sending mutated data and waiting for a response. AFL^2 extended the AFL^1 with remote monitoring strategies used in WMIFuzzer. However, the fuzzing schedule of AFL was never changed. Sulley is another state-of-the-art fuzzer that is popular for protocol fuzzing, but it needs a data model to start fuzzing. As different seeds have different fields, every seed needs a data model in Sulley. Since it is laborious, time-consuming, and error-prone to write data models by manual analysis of every seed, a data model generation that treats every message as a set of tokens separated by CR-LR or blank was used in this experimentation. In detail, parsing every seed to tokens once can generate its data model that can be integrated into the $Sulley^1$. $Sulley^2$ extended $Sulley^1$ with remote monitoring strategies used in WMIFuzzer. Anyway, the seed messages generated by WMIFUzzer are used as the initial seeds for AFL and Sulley.

Table 1: Summary of devices under testing.

Type	Vendor	Device	FirmwareAva	FirmwareDec	FirmwareRun
SOHO router	Phicomm	K2-A6	Yes	Yes	No
	JieXi	AC836M	Yes	No	No
Enterprise gateway	FeiYuXing	VE602W+	Yes	No	No
	RuiJie	NBR1300G	Yes	No	No
IP camera	RIWYTH	RW-950S	No	No	No
	NEO	NIP-25SY	No	No	No
	ZTE	C520P	Yes	Yes	No

4.2.3. Research Questions. Using the previous experiment setup, we aim to answer the following research questions:

(i) *RQ1*. Can WMIFuzzer discover vulnerabilities in COTS IoT devices based on its seed generation and fuzzing of WMPT?

(ii) *RQ2*. Can WMIFuzzer outperform the state-of-the-art fuzzers by performing mutation-based fuzzing on the same initial seeds?

4.2.4. Effectiveness of Vulnerability Detection (RQ1). Table 2 lists the unique vulnerabilities found by WMIFuzzer. For each device under testing, WMIFuzzer firstly applies the UI automation in Section 3.3 to generate seed messages in 1 hour automatically. Then, it converts the seed messages to WMPTs and starts mutation-based fuzzing in 23 hours. At last, it discovered 10 vulnerabilities: 3 command injections, 4 interface leaks, and 3 crashes.

All of them have been reported to the CNCERT/CC [21] that cooperates with many vendors to fix vulnerabilities in their products, and 6 CNVD-IDs are assigned as they are zero-days. The two crashes (Vul-ID-003 and Vul-ID-009) have been reported by other researchers independently before this paper did, so they are not been assigned with a CNVD-ID. The crash (Vul-ID-006) has been reviewed by the vendor, and they inferred that it is not security related, so it was not assigned with a CNVD-ID. Vul-ID-004 and Vul-ID-005 are different vulnerabilities in a same device, but they have been combined to be assigned with one CNVD-ID.

In addition, most of these vulnerabilities are of high impacts because they can be exploited remotely. An attacker can utilize them to compromise the targeted device, such as installing malware into a router for further cybercrimes and rebooting a camera for bypassing surveillance. At the time of this writing, patches have been released to fix these vulnerabilities.

These results indicate that WMIFuzzer can discover vulnerabilities in the web management interface of COTS IoT devices automatically based on its seed generation and mutation-based fuzzing.

4.2.5. Efficiency of Vulnerability Detection (RQ2). We measure the efficiency in terms of vulnerabilities discovered over quantity and time. Table 3 lists the results by comparing WMIFuzzer with AFL and Sulley; it shows that WMIFUzzer can detect more vulnerabilities than the other fuzzers. In detail, WMIFUzzer detected 10 vulnerabilities including *Command Injection, Interface Leak*, *and Crash* while the other two fuzzers just detected a subset. Moreover, AFL and Sulley take more time than WMIFuzzer for detecting the same vulnerability.

We performed a further manual analysis and found the following: (1) *Command Injection and Interface Leak* vulnerabilities cannot be detected by both AFL[1] and Sulley[1] because these two fuzzers can both receive response messages from the devices where no crashes are triggered. However, AFL[2] and Sulley[2] can detect them because of the integration with the remote monitoring strategies used in WMIFuzzer. (2) Sulley[2] can detect more vulnerabilities than AFL[2], and Sulley[2] also takes less time on the same vulnerability. After manual checks, we get the result that byte-level mutation in AFL[2] generated a lot of structure-invalid messages rejected by the device. In contrast, the data model in Sulley[2] can guide the mutation to generate structure-valid messages. Actually, the data model in Sulley[2] can be considered as a simple version of WMPT, where mutation on internal nodes is not permitted, every leaf node has the same weight scope, and every generated message just contains one mutated leaf node.

These results indicate that WMIFuzzer is competitive compared to the state-of-the-art fuzzers: AFL and Sulley, in terms of quantity and time about vulnerability detection in the web management interface of COTS IoT devices.

4.3. Case Study

4.3.1. Feiyuxing Enterprise Gateway. *VE602W+* is a wireless gateway designed for small office and company, and it contains web management interface. The firmware of this device has been released publicly, but it is encrypted or packed with a private format, and it cannot be unpacked by existing tools. Therefore, WMIFuzzer can be used to detect vulnerabilities in this device by blackbox fuzzing. After 30 minutes, WMIFuzzer reported a potential interface leak vulnerability as a message with mutated URL can get a successful response. This mutated URL is "http://192.168.0.1/.htpasswd," and the response content is "admin:$1$$FQzEy2IhIMAL60u9OrHLp1." After manually analyzing the content, it can be decrypted by John the Ripper password cracker [53]. The result is "admin:147258369," which is just the credential of the web management interface. So, anyone who can access the router work can get the password of this device no matter how the administrator changed his/her password. After another 4 hours, WMIFuzzer reported a potential *Command Injection* vulnerability as another message triggered the device sending a *ICMP* request to

Table 2: Summary of discovered vulnerabilities.

Device	Vul-ID	Vulnerability	Remotely exploitable	CNVD-ID	Addition
Phicomm K2-A6	001	Command injection	True	CNVD-2017-25289	Fixed
	002	Interface leak	True	CNVD-2017-20666	Fixed
JieXi AC836M	003	Crash	True	*N*-day	Fixed
FeiYuXing VE602W+	004	Interface leak	True	CNVD-2017-35720	Fixed
	005	Command injection	True		Fixed
RuiJie NBR1300G	006	Crash	False	Just-a-Dos	Fixed
	007	Command injection	True	CNVD-2018-22138	Fixed
RIWYTH RW-950S	008	Interface leak	True	CNVD-2017-37032	Fixed
NEO NIP-25SY	009	Crash	False	*N*-day	Fixed
ZTE C520P	010	Interface leak	True	CNVD-2018-21990	Fixed

Table 3: Statistics on vulnerability detection.

Vulnerability type	Device	WMIFuzzer	AFL[1]	Sulley[1]	AFL[2]	Sulley[2]
Command injection	Phicomm K2-A6	3 h 29 m	NA	NA	19 h 7 m	NA
	FeiYuXing VE602W+	5 h 37 m	NA	NA	NA	11 h 29 m
	RuiJie NBR1300G	7 h 12 m	NA	NA	NA	NA
Interface leak	Phicomm K2-A6	2 h 5 m	NA	NA	NA	40 m
	FeiYuXing VE602W+	3 h 49 m	NA	NA	18 h 52 m	10 h 21 m
	RIWYTH RW-950S	5 h 24 m	NA	NA	NA	13 h 52 m
	ZTE C520P	52 m	NA	NA	21 h 27 m	8 h 3 m
Crash	NEO NIP-25SY	18 m	2h21	25m	5 h 9 m	1 h 22 m
	JieXi AC836M	2 h 11 m	NA	NA	NA	5 h 36 m
	RuiJie NBR1300G	4 h 31 m	NA	NA	NA	NA

the fuzzer. We reviewed the mutated message content in Pseudocode 2 and found the difference that "PING_HOSTIP = 11" has been mutated to "PING_HOSTIP = 11 | ping 192.168.0.11."

Further inspection showed that the module named *System Diagnosis* is responsible for this message. This module executes *ping* command inside the device, and the parameter of *ping* is referenced to the user-supplied *PING_HOSTIP*. It seems that the developer must have forgotten to sanitize this content. So, evil content can execute additional commands.

Then, the *nmap* [54] is deployed to scan this device, and two special open ports are found: 23, that is, the *telnet* service, and 10089, that is, the SSH service. At last, the upper vulnerabilities were used to crack the device system credentials, and we succeeded in logging into the SSH service with the cracked credentials. Moreover, by manually checking the web interface, we found that there is no entry for closing these two services. This means that an attacker can utilize these vulnerabilities to control the gateway device independent of the web interface authentication.

4.3.2. Phicomm Smart Router. K2 is a smart home router that can be managed via a browser or its official app. The firmware of this device is released on the vendor website, Binwalk can unpack it, and its web management interface is implemented by Lua [33] scripts. We could not find a tool that is designed to detect vulnerabilities in Lua scripts, and we failed running the firmware in the emulator [19] because of some hardware features unavailable in the emulator. Then, WMIFuzzer was deployed to perform the blackbox fuzzing, and a crash was reported after 4 hours. Manual analyzing the crash showed that the root cause is the URL about time reboot module. This module is designed for the administrator to configure the router rebooting at a special time automatically, and the main part of the message is listed in Pseudocode 3.

Compared to the original seed message, the additional content is the "—reboot." By reviewing all unpacked Lua scripts, this crash was hosted in the script file that contains the core codes in Pseudocode 4.

This script firstly gets the request parameters named *timeRebootEnablestatus and timeRebootrange*, and then combines them to construct the parameters of *luci.sys.call* without any sanitization. So, malicious parameter in this message can execute extra commands via *luci.sys.call*.

By this *Command Injection* vulnerability, we detected the underlying architecture and found that it contains *Wget* that can download a file from a remote server. Then, *Wget* was applied to download the corresponding *Dropbear*, a popular SSH server for embedded devices, and install it into the device. Finally, we got the full control of this router independent of the web management interface authentication.

4.3.3. NEO IP Camera. In the manual, customers are suggested to manage this device via its official control app while the web management interface is also enabled. The vendor does not release the firmware publicly, and we also found

```
POST/diag.cgi HTTP/1.1
Host: 192.168.0.1
Content-Length: 82
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWeb Kit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/
537.36
Accept: text/html, application/xhtml + xml,application/xml;
q = 0.9,image/webp, image/apng,∗/∗; q = 0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh; q = 0.9
Cookie: username = %%; hash_key = 5122420728838914,session_id = 8492544218643274
Connection: close
ENABLE_EXTERNAL_PING = YES_PING_HOSTIP = 11 | ping 192.168.0.11_PING_COUNT = 3
&pg = ping&LANGUAGE = &OKBTN=Start
```

Pseudocode 2

```
—WebKitFormBoundarysrLKRlLVLplqRfAu
Content-Disposition: form-data; name = “timeRebootEnablestatus”
on | reboot
—WebKitFormBoundarysrLKRlLVLplqRfAu
Content-Disposition: form-data; name = “timeRebootrange”
00 : 00
—WebKitFormBoundarysrLKRlLVLplqRfAu
Content-Disposition: form-data; name = “cururl”
http://192.168.2.1/cgi-bin/luci/;stok=b458f583573f2e98181cf1d9ce0f8443/admin/index
—WebKitFormBoundarysrLKRlLVLplqRfAu–
```

Pseudocode 3

```
function do_timereboot()
local uci_t = require “luci.model.uci”.cursor()
local rebootenable = luci.http.formvalue(“timeRebootEnablestatus”)
local timerange = luci.http.formvalue(“timeRebootrange”)
local cururl = luci.http.formvalue(“cururl”)
luci.sys.call(“uci set timereboot.timereboot.enable = %s>/dev/null” % rebootenable)
luci.sys.call(“uci set timereboot.timereboot.time = %s>/dev/null” % timerange)
luci.sys.call(“uci commit timereboot.timereboot>/dev/null”)
. . .//other codes
end
```

Pseudocode 4

nothing from online search engines, such as Google and Baidu. So, WMIFuzzer was deployed to fuzz its web management interface automatically. After 18 minutes, a crash was reported and the corresponding mutated message was captured in Pseudocode 5.

The content of authentication header has been appended with a number of ASCII char *A*. In our whole experimentation, both AFL and Sulley can trigger this crash, but they take more time than WMIFuzzer. We manually analyzed the mutated messages in these two fuzzers and found that the crash will not be triggered if the mutation changed the token *Basic*. As AFL cannot treat *Basic* and the *YWRtaW46YWRtaW4=* as two atomic fields, its mutation rarely generates a message that keeps the first field unchanged while the second field is modified to a long string. In contrast, both Sulley and WMIFuzzer can identify these two atomic fields, so they can detect the vulnerability quickly.

5. Discussion and Limitations

Although WMIFuzzer can discover vulnerabilities in COTS IoT devices efficiently, there are still some avenues for future improvements.

5.1. Scope of Targeted Devices. This paper focuses on COTS IoT devices that have a web management interface. Although

```
GET/HTTP/1.1
Host: 192.168.100.100
Authorization: Basic YWRtaW46YWRtaW4 = AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/
537.36
Accept: text/html, application/xhtml + xml,application/xml; q = 0.9,image/webp, image/apng,
*/*; q = 0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh; q = 0.9
Connection: close
```

Pseudocode 5

this interface is very popular, not every device has it since web interface requires an underlying network that has a high bandwith (e.g., TCP/IP stack). For example, a micro temperature sensor that has a limited power resource usually applies Bluetooth Low Energy (BLE) as its network stack. WMIFuzzer cannot be used to detect vulnerabilities in these devices. However, WMPT is not dependent on web technology and any highly structured message can apply the WMPT to dissect itself for mutation-based fuzzing.

5.2. Interface Protection in Device. To detect vulnerabilities as fast as it can, WMIFuzzer sends a lot of messages to the device at a high speed. But if there is protection about message speed in the device firmware, most messages will be blocked to make WMIFuzzer inefficient. Message context is another protection, such as the CSRF token [55], which is designed to prevent CSRF attack. If this web interface contains CSRF protection, every message sent to the device must contain an unequal token extracted from the previous response. This means that a message will be invalid if it had been sent to the device once. This will make WMIFuzzer to not work since mutated messages from a seed have the same invalid CSRF token.

5.3. Limitations of Message Seed Generation. WMIFuzzer generates the initial message seeds by crawling the web pages of an IoT device. HTTP and HTTPS are very popular in transferring the web page content of IoT devices, and WMIFuzzer works well for these devices now. However, if a security scheme is deployed for this transmission (e.g., PrivHome [26]), WMIFuzzer will not work directly because of failure on decrypting captured web messages. One possible approach is implementing an internal interceptor of this new scheme and using it to capture the messages before encrypting them. A HTTP client containing this scheme is also needed for encrypting mutated messages and sending them to targeted devices. The WMPT and strategy of mutation are independent of this scheme, and they can remain unchanged.

5.4. Limitations of Remote Monitoring. As COTS IoT devices usually do not have the support for local monitoring, WMIFuzzer has to infer whether a vulnerability is triggered by monitoring the device network. But vulnerabilities in the targeted device may not alter the network status. For example, a memory overflow vulnerability may not cause a system crash if the device does not integrate MMU [56] to its memory management policy. Moreover, WMIFuzzer detects *Command Injection* dependent on *ping* and *reboot* commands since they can modify the device network status. However, if they are not available in the targeted COTS device, the device network cannot be changed by command injection vulnerabilities. More channels are required to monitor the running status of the targeted device, and one possible approach is monitoring signals from the hardware interface.

6. Related Work

As firmware images are been packed to reduce storage usage, existing tools (e.g., BinWalk, BAT, and FRAK [13, 14, 15]) are presented to unpack them. Costin et al. [5] presented the first public, large-scale analysis of firmware by analyzing the unpacked files statically. By collecting thousands of firmware images and analyzing them, the authors showed that account passwords and self-signed SSL certificates together with their corresponding private RSA keys, outdated software, building images as root and web servers configuration make many firmware images vulnerable. In another study, Costin et al. [6] used RIPS to scan PHP files in the device webroot to detect vulnerabilities, but the results showed that only 8% of embedded firmware images contain PHP code. Yan et al. [18] utilized a novel model of authentication bypass flaws to analyze binary files in firmware images, and finally they discovered several vulnerabilities in three firmware. It seems easy for static approaches to test IoT devices, as they do not need to run the firmware and they are independent of the real hardware. However, there are well-known limitations for static analysis techniques. The first one is *false negatives* (FNs), since static analysis mainly works upon some known patterns. For those vulnerabilities that have a new pattern,

static approaches usually fail. The second one is *false positives* (FPs), since every detected vulnerability by static approaches is just a potential vulnerability. A lot of manual efforts are required for further analysis to confirm them. So, some researches look forward to dynamic techniques.

Costin et al. [6] presented an automated framework to discover vulnerabilities in web interfaces of embedded devices; it works by integrating Qemu to run the web service and testing the web service via existing web penetration tools. Although it used some heuristic techniques to run *chroot* and *init* to launch the web service, it may fail because of the *side effects of forced emulation*, *diversity of web server environment*, and *limitations of Qemu*. Chen et al. [19] presented an automated dynamic analysis system for Linux-based embedded firmware; it works on building a full system emulator to run the firmware with a stocked kernel and a private library of *nonvolatile memories* (NVRAM). It overcomes the challenges of userland emulation, such as *chroot* and *init*. However, the boot and configuration information in different devices are diverse, and it is very difficult to emulate fully. So, it still requires a lot of manual efforts to model its operation of NVRAM when testing a new device. Davidson et al. [57] presented a platform FIE for discovering implementation flaws in MSP430 family of microcontrollers, and FIE works by utilizing KLEE [58] symbolic execution engine to verify security properties in open-source firmware programs. FIE can perform a complete analysis of firmware images because symbolic execution can explore all possible execution paths in a program. However, it is limited to analyzing small firmware that is open-source and written in C Zaddach et al. [20] presented another platform, Avatar, supporting dynamic analysis of firmware; it is a hybrid approach that contains both the device and an emulator integrated with the S2E [59] engine. Avatar utilizes S2E to perform selective symbolic execution in the emulator and forwards I/O operations from the emulator to the real device. Anyway, both static analysis and dynamic analysis based on firmware are dependent on the availability of device firmware images. However, firmware images of many COTS IoT devices are not available publicly, or they have been encrypted with a private key, or they have been compressed with an unknown file format.

Some other research studies applied online scanning techniques to detect embedded devices' vulnerabilities in the Internet scale. Cui et al. [60] and Cui and Stolfo [16] utilized Nmap [54] to scan a wide range of open embedded devices in the Internet and attempted to log into them using well-known default credentials. The results showed that a large number of devices ranging from enterprise equipment to office equipment are vulnerable. In another work [5], ZMap [61] was used to search vulnerable devices in the Internet based on vulnerable SSL certificates, and around 30K online affected devices were identified. Heninger et al. [62] studied vulnerable keys in embedded devices at Internet scale, and they found that 0.75% of TLS certificates share keys due to insufficient entropy during key generation. SHODAN [63] is a powerful engine to search vulnerable devices in the global network, and it works by matching open service fingerprints in online devices. However, online scanning mainly focuses on discovering known vulnerabilities, as it works on a number of known signatures. So, it usually cannot discover zero-days. On the contrary, it is also ethically questionable, even illegal, to scan online devices without authorization.

Since most network-enabled devices will communicate with an external entity, some works are presented to fuzz these communication protocols for vulnerability discovery. RPFuzzer [64] is a blackbox fuzzing framework to detect vulnerabilities in Cisco routers, and it used a predefined data model to generate seeds for mutation-based fuzzing. The main challenge is that it requires a security expert to write the data model, so it cannot be leveraged to test other devices automatically. Chen et al. [65] presented IOTFUZZER that performs a protocol-guarded fuzzing on COTS devices; its key idea is that many IoT devices can be controlled through their official mobile apps. So, they firstly adopted a taint-based approach to track the atomic data that are used to construct the network message; then, they mutated these atomic data dynamically to reuse the original code of message building. Our work is similar to IOTFUZZER; both of them perform mutation on atomic elements and do not require a predefined data model. However, not all IoT devices have an official control app, and IOTFUZZER can just detect memory corruption. In addition, the taint-based approach is much heavier than our blackbox approach. AFL and Sulley are two state-of-the-art fuzzers that are widely used to discover vulnerabilities in many software, but additional patches are needed to support fuzzing COTS IoT devices.

7. Conclusion

We have proposed the first blackbox fuzzer targeting the web management interface in COTS IoT devices; it utilizes the mutation-based fuzzing technology to discover vulnerabilities automatically. To improve the efficiency of fuzzing, a set of techniques including the seed generation, the WMPT, and the remote device monitoring are designed. The experimentation on 7 COTS devices successfully identified 10 vulnerabilities including 6 zero-days. All vulnerabilities have been reported to CNCERT/CC to help the vendor to fix them, and all of them have been fixed now.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request. The security vulnerabilities found in this paper can be accessed in the CNVD (http://www.cnvd.org.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported in part by the National Key R&D Program of China (2017YFB0802300 and 2017YFB0802900), National Natural Science Foundation of China (61602092, 61972073, and 61572115), and Open Research Project of the

State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences (2019-MS-05).

References

[1] Gartner, "Internet of things (IoT) market," 2017, https://www.gartner.com/newsroom/id/3598917.

[2] B. Douglas, "Eight crazy hacks: the worst and weirdest data breaches," 2015, https://securityintelligence.com/eight-crazy-hacks-the-worst-and-weirdest-data-breaches-of-2015/.

[3] T. Dunlap, "The 5 worst examples of IoT hacking and vulnerabilities in recorded history," 2017, https://www.iotforall.com/5-worst-iot-hacking-vulnerabilities/.

[4] S. Ravipati, "University hackers attacked 5,000 IoT devices on campus," 2017, https://campustechnology.com/articles/2017/02/13/university-hackers-attacked-5000-iot-devices-on-campus.aspx.

[5] A. Costin, J. Zaddach, A. Francillon, D. Balzarotti, and S. Antipolis, "A large-scale analysis of the security of embedded firmwares," in *Proceedings of the 23rd USENIX Security Symposium (SEC)*, pp. 95–110, Berkeley, CA, USA, August 2014.

[6] A. Costin, A. Zarras, and A. Francillon, "Automated dynamic firmware analysis at scale: a case study on embedded web interfaces," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 437–448, Xi'an, China, May 2016.

[7] 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.

[8] 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.

[9] 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.

[10] F-Secure, "Vulnerabilities in Foscam IP cameras enable root and remote control," December 2015, http://images.news.f-secure.com/Web/FSecure/%7B43df9e0d-20a8-404a-86d0-70dcca00b6e5%7D_vulnerabilities-in-foscam-IP-cameras_report.pdf.

[11] D. Roland and S. Bjarnason, "Mirai IoT botnet description and DDoS attack mitigation," 2016, https://asert.arbornetworks.com/mirai-iot-botnet-description-ddos-attack-mitigation/.

[12] DLink, "DLink firmware FTP server," 2016, http://ftp.dlink.ru/pub/.

[13] A. Cui, M. Costello, and S. Stolfo, "When firmware modifications attack: a case study of embedded exploitation," in *Proceedings of the NDSS Symposium*, San Diego, CA, USA, February 2013.

[14] H. Craig, "Binwalk: firmware analysis tool," 2010, https://code.google.com/p/binwalk/.

[15] A. Hemel, K. T. Kalleberg, R. Vermaas, and E. Dolstra, "Finding software license violations through binary code clone detection," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, pp. 63–72, Honolulu, HI, USA, May 2011.

[16] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan," in *Proceedings of the 26th Annual Computer Security Applications Conference*, pp. 97–106, Austin, TX, USA, December 2010.

[17] J. Pewny, B. Garmany, R. Gawlik, C. Rossow, and T. Holz, "Cross-architecture bug search in binary executables," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 709–724, San Jose, CA, USA, May 2015.

[18] S. Yan, R. Wang, C. Hauser, C. Kruegel, and G. Vigna, "Firmalice-automatic detection of authentication bypass vulnerabilities in binary firmware," in *Proceedings of the 2015 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February 2015.

[19] D. D. Chen, M. Woo, D. Brumley, and M. Egele, "Towards automated dynamic analysis for linux-based embedded firmware," in *Proceedings 2016 Network and Distributed System Security Symposium (NDSS)*, pp. 1–16, San Diego, CA, USA, February 2016.

[20] J. Zaddach, L. Bruno, A. Francillon, and D. Balzarotti, "Avatar: a framework to support dynamic security analysis of embedded systems' firmwares," in *Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February 2014.

[21] CNCERT/CC, *National Computer Network Emergency Response Technical Team/Coordination Center of China*, CNCERT/CC, Beijing, China, 2018, http://www.cert.org.cn/.

[22] OpenRCE, "Sulley," 2012, https://github.com/OpenRCE/sulley.

[23] M. Zalewski, "American fuzzy lop," 2014.

[24] C. Doukas, I. Maglogiannis, V. Koufi, F. Malamateniou, and G. Vassilacopoulos, "Enabling data protection through PKI encryption in IoT m-health devices," in *2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)*, Larnaca, Cyprus, November 2012.

[25] J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, "A lightweight authentication protocol for internet of things," in *Proceedings of the International Symposium on Next-Generation Electronics (ISNE)*, Tao-Yuan, Taiwan, May 2014.

[26] G. S. Poh, P. Gope, and J. Ning, "PrivHome: privacy-preserving authenticated communication in smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, 2019.

[27] H. Yang, W. Lee, and H. Lee, "IoT smart home adoption: the importance of proper level automation," *Journal of Sensors*, vol. 2018, Article ID 6464036, 11 pages, 2018.

[28] J.-H. Lee and H. Kim, "Security and privacy challenges in the internet of things [security and privacy matters]," *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 134–136, 2017.

[29] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning," 2018, http://arxiv.org/abs/1801.06275.

[30] M. Muench, S. Jan, K. Frank, A. Francillon, and D. Balzarotti, "What you corrupt is not what you crash: challenges in fuzzing embedded devices," in *Proceedings of the 2018 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, February 2018.

[31] S. Chong, J. Liu, A. C. Myers et al., "Secure web applications via automatic partitioning," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 31–44, 2007.

[32] S. Guelich, S. Gundavaram, and G. Birznieks, *CGI Programming with Perl: Creating Dynamic Web*, O'Reilly Media, Inc., Newton, MA, USA, 2000.

[33] A. Hester, R. Borges, and R. Ierusalimschy, "Building flexible and extensible web applications with Lua," *Journal of Universal Computer Science*, vol. 4, no. 9, pp. 748–762, 1998.

[34] X. Yu and Y. Cai, "Design and implementation of the website based on PHP & MYSQL," in *Proceedings of the 2010*

International Conference on E-Product E-Service and E-Entertainment, pp. 1–4, Henan, China, November 2010.

[35] A. Takanen, J. D Demott, and C. Miller, "Fuzzing for software security testing and quality assurance," 2008.

[36] C. Miller and Z. N. J. Peterson, "Analysis of mutation and generation-based fuzzing," Tech. Rep 4, Independent Security Evaluators, Baltimore, MA, USA, 2007.

[37] D. Aitel, "An introduction to SPIKE, the fuzzer creation kit," in *Proceedings of the Black Hat*, USA, August 2002.

[38] M. Eddington, "Peach fuzzing platform," 2011, https://www.peach.tech/.

[39] M. Ehmer and F. Khan, "A comparative study of white box, black box and grey box testing techniques," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 6, 2012.

[40] P. Godefroid, M. Y. Levin, and D. Molnar, "SAGE: whitebox fuzzing for security testing," *Communications of the ACM*, vol. 55, no. 3, pp. 40–44, 2012.

[41] M. Böhme, V.-T. Pham, and A. Roychoudhury, "Coverage-based greybox fuzzing as Markov chain," *IEEE Transactions on Software Engineering*, vol. 45, no. 5, pp. 489–506, 2017.

[42] F. Roy, Jim Gettys, J. Mogul et al., "Hypertext transfer protocol–HTTP/1.1," Technical Report, 1999, https://www.w3.org/Protocols/rfc2616/rfc2616.html.

[43] H. Shuai, B. Liu, S. Nath, W. G. J. Halfond, and R. Govindan, "PUMA: programmable UI-automation for large-scale dynamic analysis of mobile apps," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 204–217, Bretton Woods, NH, USA, 2014.

[44] I. Neamtiu, J. S. Foster, and M. Hicks, "Understanding source code evolution using abstract syntax tree matching," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, pp. 1–5, 2005.

[45] D. Wang, M. Jiang, T. Chen, X. Zhang, and C. Wang, "Cracking IoT device user account via brute-force attack to SMS authentication code," in *Proceedings of the First Workshop on Radical and Experiential Security*, pp. 57–60, New York, NY, USA, June 2018.

[46] W3C 2018, "HTML specification," 2018, https://www.w3.org/TR/html/.

[47] C. Willems, T. Holz, and F. Freiling, "Toward automated dynamic malware analysis using cws and box," *IEEE Security & Privacy*, vol. 5, no. 2, pp. 32–39, 2007.

[48] F. Yamaguchi, M. Lottmann, and K. Rieck, "Generalized vulnerability extrapolation using abstract syntax trees," in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 359–368, Orlando, FL, USA, December 2012.

[49] FuzzDB, "Dictionary of attack patterns and primitives for black-box application fault injection and resource discovery," 2016, https://github.com/fuzzdb-project/fuzzdb.

[50] T. F. Javier and J. D. Guttman, "Copy on write," 1995.

[51] C. McMahon, "History of a large test automation project using selenium," in *Proceedings of the 2009 Agile Conference*, pp. 363–368, Chicago, IL, USA, August 2009.

[52] mitmproxy, "An interactive TLS-Capable intercepting HTTP proxy," 2016, https://github.com/mitmproxy/mitmproxy.

[53] Solar Designer, "John the ripper password cracker," 2006, https://www.openwall.com/john/.

[54] L. Gordon, "Nmap–free security scanner for network exploration & security audits," 2009.

[55] B. Adam, C. Jackson, and J. C. Mitchell, "Robust defenses for cross-site request forgery," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 75–88, Alexandria, VA, USA, October 2008.

[56] G. Rose, "Using the microprocessor MMU for software protection in real-time systems," 2006, http://www.lynx.com/using-the-microprocessor-mmu-for-software-protection-inreal-time-systems/.

[57] D. Davidson, B. Moench, T. Ristenpart, and S. Jha, "FIE on firmware: finding vulnerabilities in embedded systems using symbolic execution," in *Proceedings of the 22nd USENIX Security Symposium (SEC)*, pp. 463–478, Washington, DC, USA, August 2013.

[58] C. Cadar, D. Dunbar, D. R. Engler et al., "KLEE: unassisted and automatic generation of high-coverage tests for complex systems programs," in *Proceedings of the OSDI*, pp. 209–224, San Diego, CA, USA, December 2008.

[59] V. Chipounov, V. Kuznetsov, and G. Candea, "S2E: a platform for in-vivo multi-path analysis of software systems," *ACM SIGPLAN Notices*, vol. 46, no. 3, pp. 265–278, 2011.

[60] A. Cui, Y. Song, P. V. Prabhu, and S. J. Stolfo, "Brave new world: pervasive insecurity of embedded network devices," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, pp. 378–380, Lecture Notes in Computer Science, Saint-Malo, France, September 2009.

[61] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: fast internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium (SEC)*, pp. 47–53, Washington, DC, USA, August 2013.

[62] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining your Ps and Qs: detection of widespread weak keys in network devices," in *Proceedings of the USENIX 21st Security Symposium*, Bellevue, WA, USA, August 2012.

[63] B. Genge and C. Enăchescu, "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services," *Security and Communication Networks*, vol. 9, no. 15, pp. 2696–2714, 2016.

[64] Z. Wang, Y. Zhang, and Q. Liu, "RPFuzzer: a framework for discovering router protocols vulnerabilities based on fuzzing," *KSII Transactions on Internet and Information System*, vol. 7, no. 8, pp. 1989–2009, 2013.

[65] J. Chen, W. Diao, Q. Zhao et al., "Iotfuzzer: discovering memory corruptions in iot through app-based fuzzing," in *Proceedings 2018 Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2018.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 9060756, 7 pages
https://doi.org/10.1155/2019/9060756

Research Article

Light Repository Blockchain System with Multisecret Sharing for Industrial Big Data

Hefeng Chen,[1] **Hsiao-Ling Wu**,[2] **Chin-Chen Chang**,[3] **and Long-Sheng Chen**[2]

[1]*Computer Engineering College, Jimei University, Xiamen 361021, China*
[2]*Department of Information Management, Chaoyang University of Technology, Taichung 413, Taiwan*
[3]*Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan*

Correspondence should be addressed to Chin-Chen Chang; alan3c@gmail.com

Received 1 March 2019; Accepted 4 August 2019; Published 16 October 2019

Guest Editor: Fagen Li

Blockchain is an emerging technology that promises many exciting applications in various fields, including financial, medical, energy, and logistics management. However, there are still some limitations in the existing blockchain framework that prevents its widespread adoption in the commercial world. One important limitation is the storage requirement, wherein each blockchain node has to store a copy of the distributed ledger. Thus, as the number of transactions increases, this storage requirement grows quadratically, eventually limiting the scalability of a blockchain system. Moreover, the public ledger in a blockchain framework allows anyone in the network to audit the transaction, which may not be favourable in some privacy-sensitive applications. In this paper, a secret-sharing scheme is proposed to reduce the size of the blockchain transactions. Each transaction block is divided into t parts, and the size of each part is $1/t$ size of transaction block. We use the secret-sharing mechanism to share t parts into n shares. Hence, each node stores not one transaction but one share in the blockchain system. The proposed scheme can eventually reduce the storage cost of a blockchain transaction by $1/t$ without introducing an additional recovery communication cost; however, robustness is reduced in node failure as a tradeoff. Meanwhile, the proposed scheme was more efficient and secure compared to other state-of-the-art schemes that aim to reduce blockchain storage for industrial big data.

1. Introduction

Internet commerce has become relatively entirely dependent on financial institutions that are trusted third parties to handle electronic payments. Although the system is suitable for most transactions, it still has the inherent weaknesses in trust-based models. Blockchain is a paradigm-shifting technology that has emerged over the past decade, which is based on peer-to-peer communication technology, network theory, and cryptography [1]. In a traditional centralized database model, the central server stores every transaction record, which can be disastrous if the central server is being attacked. In contrast, blockchain technology is a distributed solution without any third-party trust problem. Since every node in the blockchain network keeps a copy of the public ledger, it is possible to audit the transactions locally without referring to a centralized authority. Therefore, blockchain can also be viewed as a distributed ledger system, which eliminates the disadvantage of a traditional centralized database model. However, the storage cost in blockchain is huge [2], which is one of the factors currently limiting the widespread adoption of blockchain technology. Since each node needs to store the public ledger locally, the storage cost of an entire blockchain network grows quadratically.

To resolve the storage issue in blockchain, Dai et al. [3] proposed a low-storage blockchain system that employs network coding theory to divide the transaction data into multiple blocks and then stores the blocks at different nodes. This can realize distributed storage (DS) by recovering the transaction data through network coding (NC). They proposed two kinds of NC-based DS, one is deterministic rate (NC-DRDS) and the other is rateless (NC-RLDS), to deal with a fixed and variable number of blockchain nodes, respectively. In another approach, Dorri et al. [4] proposed a memory-optimized and flexible blockchain (MOF-BC) that allows the user to summarize or remove part of the "aged"

blockchain transactions. This eventually reduces the ability to perform a public audit, as some of the information is eventually erased. The idea of distributed storage blockchain (DSB) is first introduced by Raman and Varshney in 2018 [5]. The blockchain transaction block is first encrypted and then stored at different nodes (together with the encryption key) using a secret-sharing scheme. In this way, the storage of a blockchain is reduced significantly. Later, Kim et al. [6] improved DSB by proposing a local secret-sharing (LSS) scheme, which can tolerate single peer failure.

Secret sharing is used as a basic cryptographic primitive in computer science, including for electronic voting [7], key-aggregate authentication [8], distributed cloud computing [9], secret image sharing [10, 11], and data hiding [12]. In 1979, the secret sharing was first introduced by Shamir [13] based on the Lagrange interpolating polynomial, while Blakley [14] independently took another approach based on the hyperplane geometry. In 1983, Mignotte's scheme [15] and Asmuth-Bloom's scheme [16] were proposed based on the Chinese remainder theorem. A perfect (t, n) secret-sharing scheme [13] has two properties: (1) any t or more shares can recover the secret; (2) any $t-1$ or fewer shares reveal no information about the secret. To share another secret, the dealer must redistribute every participant's secret share. Later, several schemes have been presented for multiple secret sharing. In a multisecret sharing scheme, every user only needs to keep one share and many secrets can be shared independently without updating the share.

Most multisecret sharing schemes require the disclosure of large amounts of public information. Chien et al. [17] proposed a multisecret scheme based on systematic block codes, while Yang et al. [18] improved the amount of public information required by Chien et al. Further, Yang's scheme is based on Shamir's secret-sharing scheme such that fewer than the threshold number of pieces do not leak any information. Dehkordi and Mashhadi's scheme [19] focused on improving the efficiency of computations involved in share creation and secret reconstruction rather than space efficiency. Our multisecret sharing is based on recursion, in which a secret is first divided into t pieces and then the pieces are encoded one by one in such a manner that the shares of the already encoded pieces are reused to create new shares for the next piece.

In this paper, we improve existing work [3, 5, 6] by proposing a low-storage scheme with multisecret sharing based on polynomial interpolation. Our idea is different from Raman and Varshney [5] and Kim et al. [6] as we do not encrypt the blockchain transaction block. Instead, our proposed scheme divides the blockchain transaction block directly into multiple shares and stores it in different nodes. The advantages of our work can be summarized as follows.

(1) Efficiency: the size of share is on the order of $|S|/t$; that is, the storage room and communication cost are reduced effectively.

(2) Robustness: no side information needs to be stored with the shares, and no public information needs to be disclosed.

2. Blockchain and Storage Issues

Blockchain is a distributed ledger system that requires each participating node to store a copy of ledger for the transaction records. Every time a transaction block is created, it is first verified by the neighbouring nodes and then goes through the consensus (mining) process by solving a difficult cryptographic puzzle. One of the nodes that successfully solve this puzzle is rewarded with some incentives. This transaction block is then added to the ledger, wherein each block is related to the previous block through a chain of hash values. This data structure suffers from scalability issue, as the storage room required to keep the entire ledger is growing quadratically when the number of blockchain nodes increases.

3. Overview of Shamir's Secret-Sharing Scheme

3.1. Shamir's (r, n)-Threshold Secret Sharing. In 1979, Shamir developed a (r, n)-threshold secret-sharing scheme [13] based on polynomial interpolation, wherein a univariate polynomial $y = f(x)$ of degree $r-1$ is uniquely defined by r points (x_j, y_j) with distinct x_j, for $j = 1, 2, \ldots, r$. The scheme can decompose one secret into n shares, with r shares required to recover the original secret, where $r \le n$. However, the secret cannot be recovered with less than r shadows. It consists of the following two phases.

3.2. Shadow Distribution Phase. The trusted dealer starts with a secret integer, $S \ge 0$, that is, to be distributed among n users. Thus, the dealer runs the following:

(1) Choose a prime $p > \max(n, S)$.

(2) Randomly select $r-1$ independent coefficients $a_1, a_2, \ldots, a_{r-1} \in \mathbb{Z}_p$, to constitute a random polynomial with $r-1$ degree over $\mathbb{Z}_p$:

$$f(x) = S + \sum_{j=1}^{r-1} a_j x^j \bmod p. \tag{1}$$

(3) Choose n distinct nonzero elements of $\mathbb{Z}_p$, denoted as x_i, for $1 \le i \le n$.

(4) Compute $s_i = f(x_i) \bmod p$, $1 \le i \le n$ and securely transfer the share s_i to user U_i, along with the public index x_i.

3.3. Secret Reconstruction Phase. Assume that r users, $U_{i_1}, U_{i_2}, \ldots, U_{i_r}$, pool their shares to compute the secret S. Their shadows provide r distinct points $(x_{i_j}, s_{i_j})'s$, $1 \le j \le r$, which allow the computation of the coefficients of $f(x)$ by Lagrange interpolation. The secret, S, can be expressed as

$$S = f(0) = \sum_{j=1}^{r} s_{i_j} c_{i_j} \bmod p, \tag{2}$$

where $c_{i_j} = \prod_{1 \le k \le r, k \ne j} (x_{i_j} / x_{i_k} - x_{i_j}) \bmod p$, for $1 \le j \le r$.

Figure 1 shows an example of Shamir's (3, 6) secret sharing, which can be adopted by blockchain. Assume that the transaction block α is divided into three equal length packets, which are expressed as integers α_1, α_2, and α_3. We use them as the coefficients to construct a 2-degree polynomial, which is expressed as $f(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2$. The shadows are then generated and distributed to the blockchain nodes with ID = 1, 2, 3, 4, 5, 6.

4. Blockchain Transactions with Secret-Sharing Scheme

In this section, we describe our proposal of using secret-sharing scheme to reduce the storage cost of blockchain. In our sharing scheme, each block is stored in a distributed manner among all nodes. Consider the transaction block α, our idea inspired by Parakh and Kak [20]. The flowchart of sharing process and reconstruction process is shown in Figure 2.

Firstly, we divide the block α into t pieces of size $|\alpha|/t$, denoted as $\alpha_1, \alpha_2, \ldots, \alpha_t$, such that their concatenation $\alpha_1 \| \alpha_2 \| \ldots \| \alpha_t = \alpha$. All the arithmetic performed is finite field arithmetic on $\mathbb{Z}_p$, where p is larger than $\alpha_i (1 \le i \le t)$ and n. Then, we share them recursively as follows. The first step is to run Shamir's (2, 2) secret sharing for the piece α_1. That is, generate randomly a 1-degree polynomial $g_1(x)$ whose constant term is α_1, and compute $g_{1,1} = g_1(1)$ and $g_{1,2} = g_1(2)$ to obtain two shares for α_1. Next, generate a 2-degree polynomial $g_2(x) = \alpha_2 + g_{1,1}x + g_{1,2}x^2 \pmod p$ whose constant term is the next piece α_2, and the coefficients are the previous two shares. By induction, assume that we have obtained i shares of the piece $\alpha_{i-1} (2 \le i \le t-1)$, denoted as $g_{i-1,1}, g_{i-1,2}, \ldots, g_{i-1,i}$, we can generate a i-degree sharing polynomial for the next piece α_i by using $g_{i-1,j}$ as the j-th coefficient term for $j = 1, 2, \ldots, i$. At the last step, we generate a t-degree sharing polynomial:

$$g_t(x) = \alpha_t + g_{t-1,1}x + g_{t-1,2}x^2 + \cdots + g_{t-1,k}x^t \pmod p, \tag{3}$$

and the final n shares are

$$g_t(x_1), g_t(x_2), \ldots, g_t(x_n), \tag{4}$$

where $x_1, x_2, \ldots, x_n$ are n public indexes of nodes. The sharing scheme is shown in Algorithm 1.

The reconstruction is an inverse process that is carried out in a backward and first-out manner. Any $t+1$ of the shares can interpolate a t-degree polynomial $g_t(\cdot)$ with the constant term α_t. Then, interpolate a $t-1$-degree polynomial $g_{t-1}(\cdot)$ with the constant term α_{t-1}, by taking the j-th coefficient of $g_t(\cdot)$ as the point at $x = j$, for $j = 1, 2, \ldots, t$. The recursion repeats until α_1 is obtained. Then, the whole block is given by the concatenation $\alpha_1 \| \alpha_2 \| \ldots \| \alpha_t = \alpha$. Algorithm 2 shows how to reconstruct the shared transaction block.

In Figure 3, a block α is divided into 4 pieces $\alpha_1 = 19$, $\alpha_2 = 27$, $\alpha_3 = 4$, and $\alpha_4 = 15$. Assumed that α is to be shared between 7 parties such that any 5 of them can reconstruct all the 4 secrets. We can now use a prime $p = 31$ and random number $r = 11$.

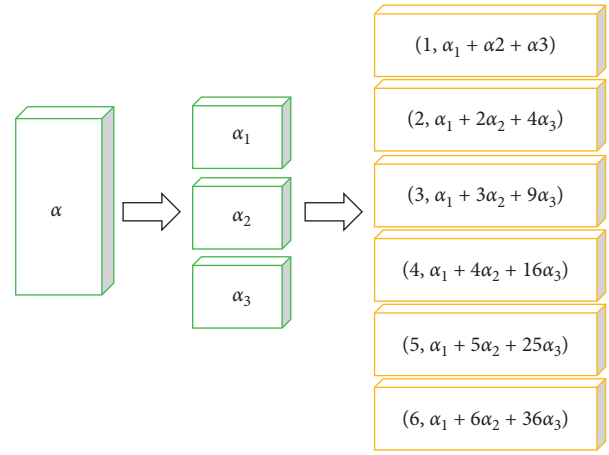


Figure 1: Optional (3, 6) Shamir secret sharing.

Theorem 1. *Our proposed scheme divides the original file α into n pieces in such a way that (i) any $t+1$ packets can recover the original file α, but (ii) no group of t pieces can do so.*

Proof. Based on the Lagrange interpolation, given t points (u_{i_1}, y_{i_1}), (u_{i_2}, y_{i_2}), ..., $(u_{i_{t+1}}, y_{i_{t+1}})$ with distinct u_{i_j}'s, there is a unique polynomial of degree t such that $y_{i_j} = g(u_{i_j})$, for $1 \le j \le t+1$, i.e.,

$$g(x) = \sum_{j=1}^{t+1} y_{i_j} l_{i_j}(x) \bmod p, \tag{5}$$

where $l_{i_j}(x) = \prod_{1 \le k \le t+1, i_k \ne i_j} (x - u_{i_k}/u_{i_j} - u_{i_k}) \bmod p$.

Therefore, $\alpha_t = g(0)$ can be reconstructed correctly. To reconstruct the next piece α_{t-1}, a $(t-1)$-degree polynomial is generated by taking the k-th coefficient of $g(x)$ as the point at $x = k$, for $k = 1, 2, \ldots, t$. Correctness is still guaranteed by Lagrange interpolation. Then, based on recursion, α_{t-2}, ..., α_2, α_1 can be correctly reconstructed one by one.

On the contrary, if only t of these n packets are available, it is not suffice to compute α_i's. For each candidate value $\alpha_t \in \{0, 1, \ldots, p-1\}$, we can construct a unique polynomial $g(x)$ of degree t such that $g(0) = \alpha_t \bmod p$ and $g(x_{i_j}) = y_{i_j} \bmod p$ for $1 \le j \le t+1$. These p possible polynomials have equal probability; therefore, the real value of packets cannot be determined.

Moreover, we can improve the above proposed scheme to achieve the verifiable property. In general, each transaction block in the blockchain system (e.g., Bitcoin) consists of a block header and a data block. All transactions are included in the data block as leaf nodes of a Merkle tree [21], and the hash root of the Merkle tree is computed in the block header (see Figure 4). Denote the block header and data block of a transaction block, α, as β and δ, respectively. Now, we perform the above sharing scheme for β and δ, independently. After reconstructing β and δ, the correctness of δ can be verified by the root hash value in block header β. □

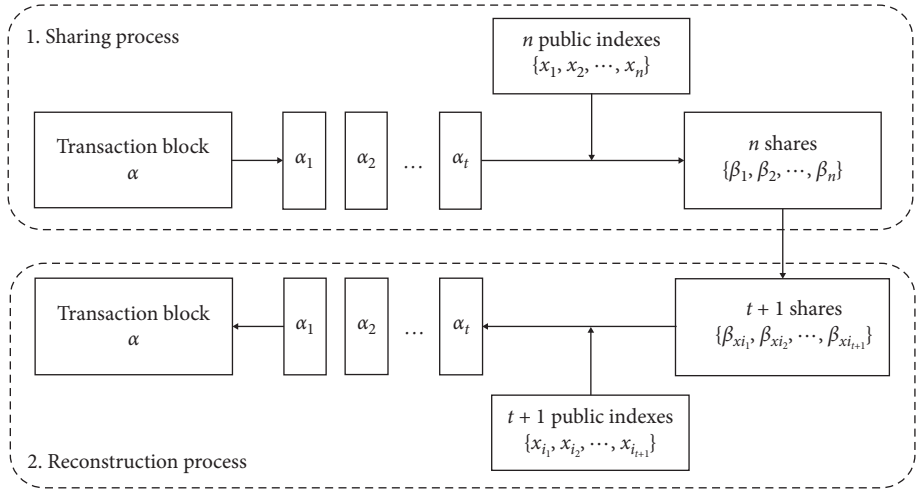


Figure 2: Flowchart of sharing process and reconstruction process.

Given the block α and n public indexes $\{x_1, x_2, \ldots, x_n\}$
  Divide α into t equal length pieces $\alpha_1, \alpha_2, \ldots, \alpha_t$
  Select a large prime $p > \max\{\alpha_1, \alpha_2, \ldots, \alpha_t, n\}$
  Select a random number $r \in \mathbb{Z}_p$, and construct a 1-degree polynomial $g_1(x) = \alpha_1 + rx \pmod p$
  Compute $g_{1,1} = g_1(1)$ and $g_{1,2} = g_1(2)$
  for $i = 2$ to $t - 1$ **do**
    Construct i-degree polynomial $g_i(x) = \alpha_i + g_{i-1,1}x + g_{i-1,2}x^2 + \cdots + g_{i-1,i}x^i \pmod p$
    for $j = 1$ to $i + 1$ **do**
      Compute $g_{i,j} = g_i(j)$
    end for
  end for
  Construct t-degree polynomial $g_t(x) = \alpha_t + g_{t-1,1}x + g_{t-1,2}x^2 + \cdots + g_{t-1,k}x^t \pmod p$
  for $i = 1$ to n **do**
    Compute $\beta_i = g_t(x_i)$
  end for
  Distribute the share β_i to the corresponding node with public index x_i, for $i = 1, \ldots, n$

Algorithm 1: Sharing process.

Given any $t + 1$ shares $\beta_{x_1}, \beta_{x_2}, \ldots, \beta_{x_{t+1}}$
  Interpolate $t + 1$ points, (x_1, β_{x_1}), (x_2, β_{x_2}), $\ldots$, $(x_{t+1}, \beta_{x_{t+1}})$, to generate t-degree polynomial
  $g_t(x) = \alpha_t + g_{t-1,1}x + g_{t-1,2}x^2 + \cdots + g_{t-1,k}x^t \pmod p$
  Compute the constant term $\alpha_t = g_t(0)$ and the coefficients
  for $i = k - 1$ to 1 **do**
  Interpolate $i + 1$ points $(1, g_{i,1})$, $(2, g_{i,2})$, $\ldots$, $(i + 1, g_{i,i+1})$ to generate i-degree polynomial
  $g_i(x) = \alpha_i + g_{i-1,1}x + g_{i-1,2}x^2 + \cdots + g_{i-1,i}x^i \pmod p$
  Compute $\alpha_i = g_i(0) \pmod p$
end for
return $\alpha = \alpha_1 \| \alpha_2 \| \ldots \| \alpha_t$

Algorithm 2: Reconstruction process.

5. Practical Implementation Issues

5.1. Storage Cost. Assuming that the block header is stored in $\mathbb{F}_\eta$ and the data block is stored in $\mathbb{F}_q$, then the blockchain's storage cost per transaction per peer is

$$\log_2 \eta + \log_2 q. \tag{6}$$

The storage cost of the DSB proposed by Raman and Varshney [5] is

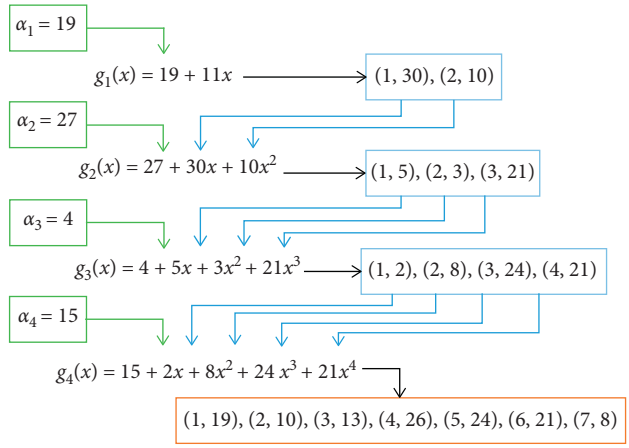


Figure 3: Example of proposed (5, 7) secret sharing where the block $\alpha = 19270415$, the prime modules $p = 31$, and the random number $r = 11$.

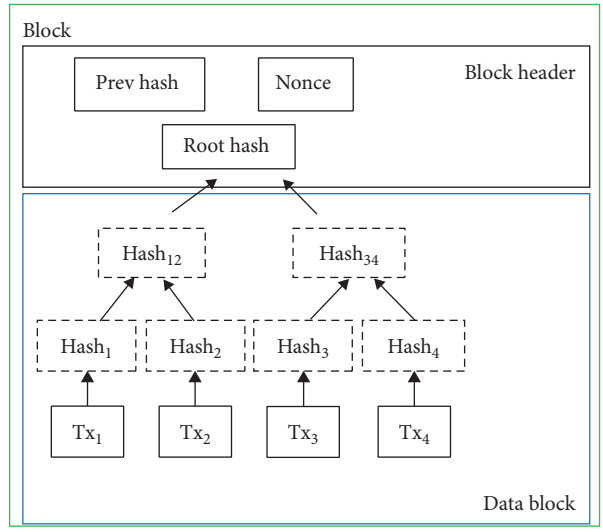


Figure 4: Transactions hashed in a Merkle tree.

$$\frac{\log_2 \eta}{t+1} + 2\log_2 q. \tag{7}$$

This is effectively smaller compared to the traditional blockchain. Kim et al. [6] proposed the LSS scheme to further reduce the storage cost to

$$\frac{\log_2 \eta}{t} + \log_2 q. \tag{8}$$

The comparison of storage cost for different schemes is detailed in Table 1.

Compared to the traditional blockchain, our proposed scheme can effectively reduce the storage cost by $1/t$, so the storage cost becomes $(\log_2 \eta + \log_2 q)/t$. From Table 1, it is clear that our proposed scheme is more efficient than other prior work including Kim et al.'s [6]. This can effectively reduce the deployment and maintenance cost in blockchain.

5.2. Recovery Communication Cost. Our proposed scheme can effectively reduce the storage cost in blockchain communication, but it also introduces some additional communication cost. In traditional blockchain, each node keeps a public ledger locally. If any node failure happens, the failed node can recover the ledger by getting it from other peers, so the communication cost is proportional to the storage cost. The minimum communication cost is essentially

$$\log_2 \eta + \log_2 q. \tag{9}$$

When there is a node failure, our proposed scheme requires $(t+1)$ secret shares to recover the transaction block, so the recovery communication cost is dependent on the number of shares $(t+1)$ and proportional to the storage cost:

$$\frac{(t+1)(\log_2 \eta + \log_2 q)}{t+1} = \log_2 \eta + \log_2 q. \tag{10}$$

Figure 5 shows the tradeoff between storage and recovery communication cost for a specific case. It is clearly shown that our proposed scheme can achieve lower storage cost but

Table 1: Comparison of storage/communication costs and robustness.

	Traditional blockchain	Original DSB [5]	DSB with LSS [6]	Proposed scheme
Storage cost	$\log_2 \eta + \log_2 q$	$(\log_2 \eta/t + 1) + 2\log_2 q$	$(\log_2 \eta/t) + \log_2 q$	$(\log_2 \eta + \log_2 q)/t$
Communication cost	$\log_2 \eta + \log_2 q$	$\log_2 \eta + 2(t+1)\log_2 q + \rho$	$\log_2 \eta + t\log_2 q$	$\log_2 \eta + \log_2 q$
Robustness	$n-1$	$(n/t+1)-1$	$(2n/t+1)-1$	$n-t-1$

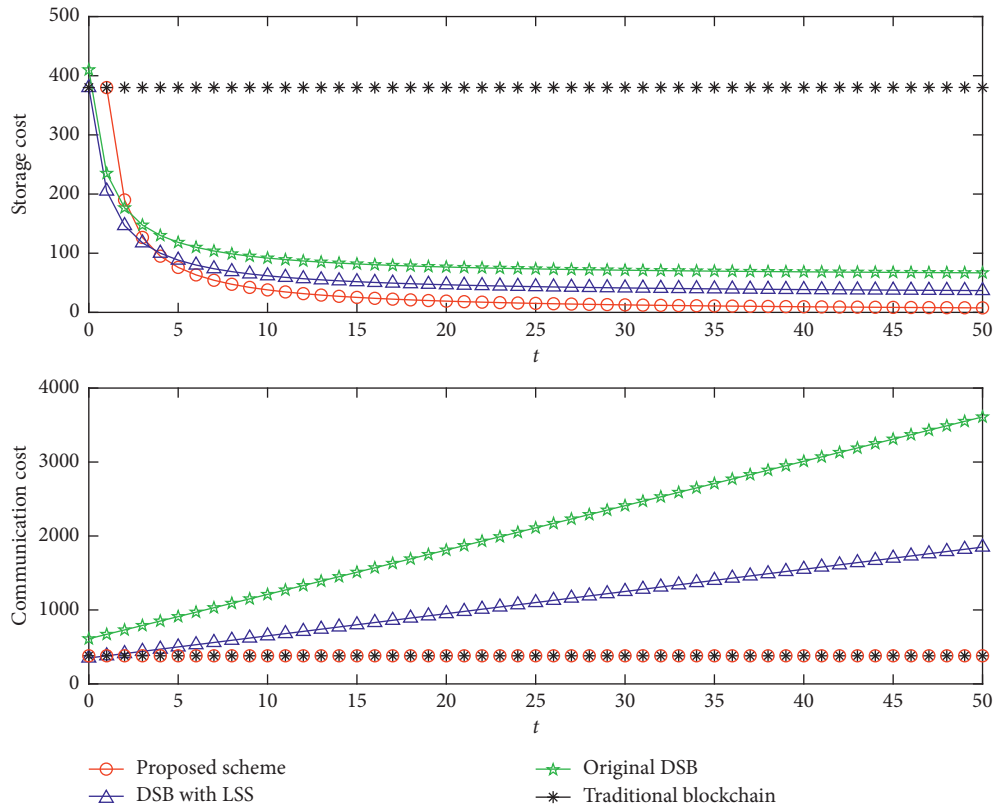


Figure 5: Tradeoff between storage cost and recovery communication cost ($\eta = 2^{350}$, $q = 2^{30}$, and $\rho = 200$).

does not increase the recovery communication cost. This is a huge advantage compared to original blockchain and the state-of-the-art work by Kim et al.

5.3. Robustness to Peer Failures. In regard to the robustness to peer failures, traditional blockchain can tolerate $(n-1)$ node failure, in the expense of large ledger stored in every node. The scheme proposed by Raman and Varshney [5] and Kim et al. [6] can only tolerate $(n/t+1)-1$ and $2(n/t+1)-1$ failures, respectively, which is significantly smaller than the traditional blockchain. Our proposed scheme can tolerate $(n-t-1)$ node failure, which is better compared to these two state-of-the-art schemes [5, 6]. Figure 6 shows the comparison of robustness where $n = 900$.

Another interesting work was presented by Dai et al. [3] recently based on network coding theory. Compared to them, our proposed scheme shares the same storage cost, communication cost, and robustness. However, their scheme can guarantee the recovery of transaction block if the collected packets is no less than t. It may leak information to malicious node if the collected packets is less than t. On the contrary, our proposed scheme utilizes Shamir's secret-sharing scheme, which can assure that if the collected packets is less than r, no additional information is being leaked. Hence, it is more advantageous compared to the scheme proposed by Dai et al.

6. Conclusions

Traditional blockchain requires huge storage room when the number of nodes increases, which limits the scalability of this emerging technology. In this article, we proposed a low-storage scheme based on secret sharing to reduce the storage cost of blockchain. The proposed scheme is able to reduce the storage cost of traditional blockchain by $1/t$ without introducing additional communication cost. It is also more efficient compared to the other recent studies [5, 6] that attempted to solve the same problem. Although the scheme proposed by Dai et al. [3] is as efficient as our proposed scheme, it leaks partial information and is considered less secure. The proposed scheme requires both r and n to be fixed, so it is more suitable for private and consortium blockchain systems where the number of nodes is

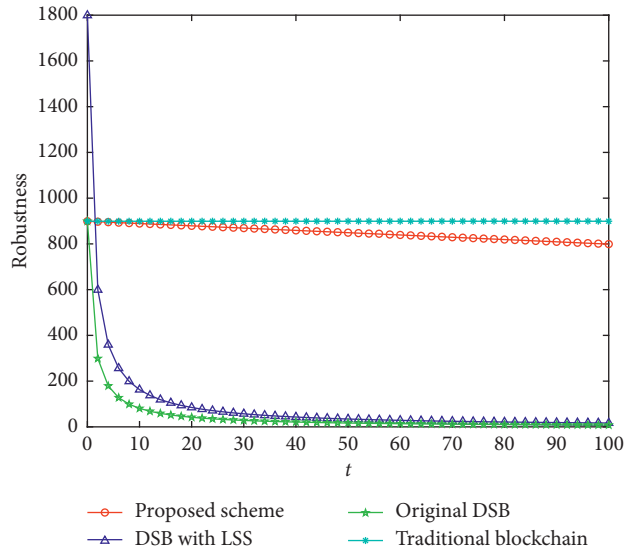


Figure 6: Comparison of robustness ($n = 900$).

predetermined. This can be a limitation as the number of nodes in public blockchain is changing dynamically. In future, we plan to extend this to support public blockchain.

Data Availability

The relevant analysis data used to support the findings of this study are included in the article.

Conflicts of Interest

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

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (Grant no. 11804114), the Natural Science Foundation of Fujian Province, China (Grant nos. 2017J01761 and 2018J01537), the Science and Technology project of Fujian Province (Grant no. 2019H0021), and the Science and Technology project of Xiamen Municipal (Grant no. 3502Z20173028).

References

[1] V. Sharma, "An energy-efficient transaction model for the blockchain-enabled Internet of Vehicles (IoV)," *IEEE Communications Letters*, vol. 23, no. 2, pp. 246–249, 2019.

[2] K. Croman, C. Decker, I. Eyal et al., "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, pp. 106–125, Springer, Berlin, Germany, 2016.

[3] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22970–22975, 2018.

[4] A. Dorri, S. S. Kanhere, and R. Jurdak, "MOF-BC: a memory optimized and flexible blockchain for large scale networks," *Future Generation Computer Systems*, vol. 92, pp. 357–373, 2019.

[5] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *Proceedings of Information Theory and Applications Workshop (ITA)*, San Diego, CA, USA, February 2018.

[6] Y. Kim, R. K. Raman, Y.-S. Kim, L. R. Varshney, and N. R. Shanbhag, "Efficient local secret sharing for distributed blockchain systems," *IEEE Communications Letters*, vol. 23, no. 2, pp. 282–285, 2018.

[7] S. Iftene, "General secret sharing based on the Chinese remainder theorem with applications in E-voting," *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67–84, 2007.

[8] C. Guo, N. Luo, M. Z. A. Bhuiyan et al., "Key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage," *Future Generation Computer Systems*, vol. 84, pp. 190–199, 2018.

[9] H. Pilaram and T. Eghlidos, "An efficient lattice based multi-stage secret sharing scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 2–8, 2015.

[10] C. Guo, Q. Yuan, K. Lu, M. Li, and Z. Fu, "(t, n) Threshold secret image sharing scheme with adversary structure," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21193–21210, 2017.

[11] C. Guo, H. Zhang, Z. Fu, B. Feng, and M. Li, "A novel proactive secret image sharing scheme based on LISS," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19569–19590, 2018.

[12] P. Singh and B. Raman, "Reversible data hiding based on Shamir's secret sharing for color images over cloud," *Information Sciences*, vol. 422, pp. 77–97, 2018.

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[14] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the 1979 AFIPS National Computer Conference*, vol. 48, pp. 313–317, New York, NY, USA, June 1979.

[15] M. Mignotte, "How to share a secret," in *Proceedings of the Workshop on Cryptography, EUROCRYPT*, pp. 371–375, Burg Feuerstein, Germany, April 1982.

[16] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 208–210, 1983.

[17] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 12, pp. 2762–2765, 2000.

[18] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A (t, n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.

[19] M. H. Dehkordi and S. Mashhadi, "New efficient and practical verifiable multi-secret sharing schemes," *Information Sciences*, vol. 178, no. 9, pp. 2262–2274, 2008.

[20] A. Parakh and S. Kak, "Space efficient secret sharing for implicit data security," *Information Sciences*, vol. 181, no. 2, pp. 335–341, 2011.

[21] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008, https://bitcoin.org/en/bitcoin-paper.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 1283495, 9 pages
https://doi.org/10.1155/2019/1283495

Research Article

A Neighbor Prototype Selection Method Based on CCHPSO for Intrusion Detection

Yanping Shen,[1,2] **Kangfeng Zheng**,[1] **Chunhua Wu**,[1] **and Yixian Yang**[1]

[1]*School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*School of Information Engineering, Institute of Disaster Prevention, Langfang 101601, China*

Correspondence should be addressed to Yanping Shen; shenyanping@cidp.edu.cn

Received 12 March 2019; Accepted 4 August 2019; Published 20 August 2019

Guest Editor: Fagen Li

Nearest neighbor (NN) models play an important role in the intrusion detection system (IDS). However, with the advent of the era of big data, the NN model has the disadvantages of low efficiency, noise sensitivity, and high storage requirement. This paper presents a neighbor prototype selection method based on CCHPSO for intrusion detection. In the model, the prototype selection and feature weight adjustment are performed simultaneously and k-nearest neighbor (KNN) is used as the basic classifier. To deal with large-scale optimization problems, a cooperative coevolving algorithm based on hybrid standard particle swarm and binary particle swarm optimization, which employs the divide-and-conquer strategy, is proposed in this paper. Meanwhile, a fitness function based on the accuracy and data reduction rate is defined in the CCHPSO to obtain a set of appropriate prototypes and feature weights. The KDD99 and NSL datasets are used to assess the effectiveness of the method. The empirical results indicate that the data reduction rate of the proposed method is very high, ranging from 82.32% to 92.01%. Compared with all the data used, the proposed method can not only achieve comparable accuracy performance but also save a lot of storage and computing resources.

1. Introduction

With the continuous development of technology and scale of network, network security incidents have been frequent and the cyber security has already become the focus of all countries in the world. Thousands of companies and agencies around the world were attacked by a ransomware called WannaCry in 2017. The WannaCry has been harmful to 200 thousand computers in more than 150 countries [1]. The attack has had an impact on a large number of institutions around the world. It is necessary to adopt appropriate security technologies such as encryption technology, authentication technology, firewall technology, and antivirus technology [2, 3]. Only under the firewall and the user identity authentication system can not guarantee the cyber security. The intrusion detection system (IDS), as the second security line of active security protection technology, has always been the favor of the researchers.

In different detection environments, IDS can be divided into network intrusion detection system (NIDS) and host intrusion detection system (HIDS). The NIDS employs the network traffic as its data source, and the data source of the HIDS comes from the audit log of the system. Nowadays, the application of machine learning into intrusion detection system has been a trend. There are many intrusion detection systems based on k-nearest neighbors (KNNs), support vector machines (SVMs), extreme learning machines (ELMs), naive Bayes (NBs), decision trees (DTs), and so on. As one of the ten classical algorithms in the field of data mining, KNN is a lazy learning- and instance-based method. Because of the advantages of simple theory, easy implementation, and no need for pretraining, it has been widely used in the field of intrusion detection [4–12]. However, KNN suffers from two major drawbacks [13]. Firstly, the computational complexity and storage consumption are high. Secondly, the algorithm is sensitive to noise samples and isolated objects. With the expansion of network, the network traffic is increasing exponentially. A number of redundant and noise variables exist, which affects the efficiency and accuracy of the detection model [14, 15].

Therefore, there is an urgent need for data reduction techniques.

Data reduction or prototype reduction can be realized by prototype selection (PS) or prototype generation (PG). NN models can be used to guide the search for PS and PG techniques. Prototype selection (PS) technique refers to how to select a set of prototypes from the original training dataset that can represent the training dataset. A minimal set of prototypes can be obtained after prototype selection, so that the performance of a NN model trained on the prototypes is approximately as well as or better than that of a NN model built on the original dataset. PS involves identifying the best subset of the original dataset, and PG concerns creating a new set of objects which can represent the original one. Like feature selection, PS and PG can also be divided into the filter and wrapper [14]. In the filter method, only part of the training dataset is used in the progress of evaluation, while the wrapper method relies on the complete training dataset. The wrapper method can get more higher accuracy although it is more computationally expensive.

The prototype selection mechanisms include condensation, edition, and hybrid [14]. The condensation method is designed to retain samples closer to the decision boundary. The internal points do not affect decision boundaries like boundary points, so internal points can be deleted if the impact on the classification is relatively small. This method can maintain the accuracy of the training set, but the generalization accuracy of the test set may be affected. Since there are fewer boundary points than interior points in most datasets, the condensation method usually has higher compression capabilities. The edition method tries instead to remove the boundary point and keep a smoother decision boundary. The data reduction rate of this method is low. The hybrid approach removes internal and boundary points according to certain criteria and attempts to find a minimal subset that maintains or even increases the precision of generalization in the test dataset. The search directions for prototype selection include incremental, decremental, batch, mixed, and fixed [14]. For the incremental strategy, the size of the selected prototype subset gradually increases from the empty subset. The decremental strategy is just the opposite, and samples that do not meet the standard are gradually deleted. However, it suffers from high complexity over incremental algorithms. The batch method removes all instances that do not meet the criterion at once. The mixed search can iteratively add or remove the instances that meet the criterion. The fixed search is a special case of the mixed strategy. The size of the selected prototypes is fixed, i.e., the number of the additions or removals remains the same.

Most of the prototype selection techniques are combined with 1-NN, mainly because 1-NN is more sensitive to noise samples. This paper also uses 1-NN as the base learner. The heuristic intelligence method for prototype selection has excellent performance both in accuracy and reduction rate. It can improve the classification accuracy of 1-NN and reduce the data by 90% or more [16]. Therefore, researchers have studied combinatorial methods based on heuristic intelligence and nearest neighbor classification [17–24]. In this paper, we use the wrapper method to select prototypes and apply the hybrid selection method and mixed search strategy for prototype reduction. As the feature weighting can enhance the performance of KNN and the feature and instance selection are closely related [25], the feature weighting and prototype selection are simultaneously optimized in this paper. The swarm intelligence heuristic algorithm is a good scheme to do this job [22]. Swarm intelligence (SI), first proposed in 1993, is inspired by animal behaviors such as birds, ants, and fish and is a branch of a population-based heuristic method. SI algorithm has black box optimization capability and does not require prior knowledge of the required field. Particle swarm optimization (PSO), an effective algorithm in swarm intelligence, is commonly used because of its less parameter adjustment and easy implementation.

This paper proposes a method of combining the prototype selection and feature weighting adjustment. We first choose the initial prototypes using the stratification strategy which ensures that every class at least has a prototype as the representative. Then the prototype selection and feature weighting can be combined to improve the performance of KNN. This is obviously an optimization problem and can be solved by the swarm intelligence algorithm. However, with the increase in the dimension of the problem, the performance of many swarm intelligent algorithms will be poor. Thus, a cooperative coevolutionary framework, CCHPSO based on hybrid standard particle swarm and binary particle swarm optimization, is proposed in this paper. It adopts a divide-and-conquer strategy, which can deal with large-scale optimization problems. Finally, two public datasets are used to evaluate the performance of the proposed approach. Experimental results show that the framework of using the prototype selection method gives comparable accuracy than that of using all datasets. This method can also save a lot of storage and computing resources which has a wide range of application prospect in the era of big data.

The paper is organized as follows: Section 2 gives the related works. The background techniques are listed in Section 3. Section 4 reports the method this paper proposed. The experimental results are presented in Section 5. Finally, some concluding remarks are given in Section 6.

2. Related Work

Because of its key advantage of simplicity and high precision, the KNN model and its variants have been widely used in the field of intrusion detection. Aburomman and Ibne Reaz [4] combined six k-nearest neighbor (KNN) models and six support vector machine (SVM) experts using PSO. They showed that the method has better accuracy than weighted majority voting (WMV). Meng et al. [5] proposed an enhanced filter method of misuse intrusion detection, and KNN is adopted as the false alarm filter. They showed the performance of the signature-based IDS has been enhanced. Meng et al. [6] developed an alert verification, and KNN is used to filter out unwanted alarms. They showed the alarm filter can effectively filter out plenty of alarms. Tsai and Lin [7] proposed a method named "TANN." The training dataset is divided into five categories by k-means, and new features

of training dataset are formed by the area of the triangle which connects any two cluster centers and one of the original training samples. Finally, the KNN classifier is used to detect attacks based on the new dataset. Lin et al. [8] presented the CANN model which is also a new feature representation approach. It is worth mentioning that KNN is also selected to do the final classification. The above two papers give new feature representation approaches, and KNN is used as a benchmark for all the other classifiers. Sharafaldin et al. [9] produced a new type of network data which includes normal type and seven types of attacks. The machine learning algorithms were evaluated over the dataset and they reported that the KNN, random forest, and ID3 have good performance. Kuttranont et al. [10] showed that KNN is one of the promising approaches. Since big data exerts great pressure on machine learning algorithms, they proposed the implementation of KNN on GPU. Chen et al. [11] proposed a compressed model using MapReduce. KNN and SVM are employed to evaluate the performance of the compressed model. KNN has been widely applied in the above works; however, the prototype selection under the guidance of KNN is not considered.

There are many methods proposed about the prototype selection or prototype generation. Most of them use divide-and-conquer and merging strategies to select or generate new artificial samples. Haro-García and García-Pedrajas [26] proposed a divide-and-conquer recursive approach for very large problems. The method divides the original training dataset into small subsets where the prototype selection is applied. Then, the selected prototypes are rejoined in a new training dataset, and the above procedure is repeated again. Triguero et al. [27] developed a MapReduce-based framework named MRPR to distribute the functioning of the prototype reduction algorithms. The authors offer a MapReduce paradigm that gives a simple and efficient environment to parallelize the prototype reduction computation. How to produce the prototypes is not the focus of this article. Escalante et al. [28] introduced a novel approach named PGGP of PG methods. Highly effective prototypes are built based on genetic programming in which many training samples are combined through arithmetic operators. The authors showed that the method outperforms other PG approaches. Paredes and Vidal [29] proposed a new gradient descent method named learning prototype and distance (LDP). A small number of prototypes are selected, and then the position of the prototypes and their weights have been iteratively adjusted.

Some heuristic algorithms have been applied to prototype selection or prototype generation. Nanni and Lumini [18] proposed a prototype reduction method based on particle swarm optimization. The algorithm flow is similar to the processing of the random subspace in the random forest. During the training phase, the prototype generation is repeated many times, then each of the training model is used to classify each test sample, and finally the classification results are combined by the majority vote rule. Triguero et al. [19] reported a prototype generation methodology about positioning adjustment. Differential evolution is used to optimize the positioning of prototypes in nearest neighbor classification. Rezaei and Nezamabadi-Pour [20] applied the gravitational search algorithm (GSA) to generate prototypes for nearest neighbor classification. The initial objects are extracted using the stratification strategy. Derrac et al. [21] presented an approach which integrates instance selection, feature weighting, and instance weighting schemes into one. They reported that the approach can enhance the results of the 1-NN classifier. Pérez-Rodríguez et al. [22] proposed a framework of combining instance and feature selection and weighting to improve the performance of the data mining methods. Differential evolution and a binary CHC genetic algorithm are adopted to perform the weighting adjustment and selection, and 1-NN is used as the classifier. Escalante et al. [23] introduced a multiobjective evolutionary algorithm based on NSGA-II for prototype generation. Kardan et al. [24] proposed a novel hybrid approach named BBO-KNN. The biogeography-based optimization (BBO) is used to optimize the input features, feature weight, and parameter K of KNN rule.

3. Background

3.1. k-Nearest Neighbor (KNN). k-Nearest neighbor (KNN) is a simple and effective classification technique. Unlike SVM, KNN can directly deal with multiclass problems and has a wide range of applications.

KNN is a supervised classification algorithm. The training samples are expressed as (x_i, y_i), where $x_i = (x_{i1}, x_{i2}, \ldots, x_{i\,D}) \in R^D$, D represents the number of features, and y_i represents the label. For a test sample, its label will be determined by its peripheral training samples, that is, it will be predicted by the majority of the labels of the training samples around it. Generally, Euclidean distance is used to measure the similarity between the samples, which is defined as follows:

$$d\left(x_i, x_j\right) = \sqrt{\sum_{r=1}^{D} \left(x_{ir} - x_{jr}\right)^2}, \tag{1}$$

where $d(x_i, x_j)$ denotes the Euclidean distance between x_i and x_j and x_{ir} represents the r-th feature of the i-th sample.

The parameter K in KNN represents the number of neighbor samples around the query sample, and the selection of K is important for the performance of the KNN.

3.2. Prototype Selection. Prototype selection, as a data preprocessing step, can remove the noise and abnormal points and reduce the size of the training set. Let TR represents the original training dataset (including the noise and redundant information). Select TS from TR whose size is less than that of TR, yet the accuracy based on TS is almost the same as that based on TR. TS takes the place of TR as the benchmark data for training, thus saving the storage space and reducing the computational complexity.

3.3. Particle Swarm Optimization. In PSO, every particle in a D-dimensional space represents a potential solution. The particle has two properties, including the velocity and

Input: the algorithm parameters
Output: the global optimal result.
(1) **Repeat**
(2) **for** each swarm j
(3) **for** each particle i
(4) **If** $f(b(j, P_j \cdot x_i)) < f(b(j, P_j \cdot y_i))$ **then** $P_j \cdot y_i = P_j \cdot x_i$ **end if**;
(5) **If** $f(b(j, P_j \cdot y_i)) < f(b(j, P_j \cdot y'))$ **then** $P_j \cdot y' = P_j \cdot y_i$ **end if**;
(6) **end for**
(7) Perform the position and velocity update using (2), (3), or (4)
(8) **end for**
(9) **until** termination is met;

Algorithm 1: The CPSO-S_K: for a particle I, $P_j \cdot x_i$, $P_j \cdot y_i$, and $P_j \cdot y'$ denote the position, the personal best position, and the global best particle of the swarm P_j, f represents the fitness function, and b (j, z) return the global solution where the j-th position is z.

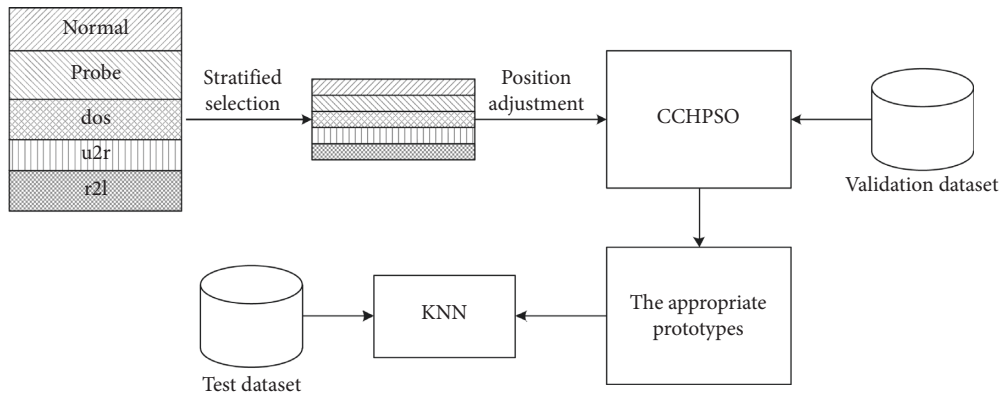


Figure 1: Architecture of the proposed model for intrusion detection.

position. The fitness is also an important property which is the evaluation of a particle. The optimal position ($p$best) and the global optimal position ($g$best) can be simultaneously perceived by every particle. The velocity and position are updated as follows:

$$V_i^t = \omega_{\text{pso}} \times V_i^{t-1} + c_1 \times \text{rand}(\,) \times \left(p\text{best}_i^t - X_i^t\right) + c_2 \times \text{rand}(\,) \times \left(g\text{best}_i^t - X_i^t\right), \quad (2)$$

$$X_i^t = X_i^{t-1} + V_i^{t-1}, \quad (3)$$

where V_i^t and X_i^t indicate the velocity and position of the i-th particle in the t-th iteration, $p\text{best}_i^t$ and $g\text{best}_i^t$ represent the previous best position of the i-th particle and the global optimal position until iteration t, w_{pso} is the constriction coefficient, c_1 and c_2 are acceleration coefficients, and rand( ) is a random number which is uniformly distributed in [0, 1].

The discrete binary version of PSO (BPSO) was designed by Kennedy and Eberhart [30]. In BPSO, the position is made of a binary string. Compared with the standard particle swarm, only the position update rules are different which is as follows:

$$X_i^t = \begin{cases} 1, & \text{if rand}(\,) \le s\left(V_i^t\right), \\ 0, & \text{otherwise}, \end{cases} \quad (4)$$

where V_i^t is mapped to interval [0, 1] by sigmoid function $s(V_i^t) = 1/(1 + e^{-V_i^t})$.

To solve large-scale optimization problems, a cooperatively coevolving PSO, CPSO-S_K, was proposed by VandenBergh and Engelbrecht [31]. The idea is very simple, and the divide-and-conquer strategy is employed. The solution can be split into L subcomponents, and each will evolve in the pattern of the PSO. The final global optimal position is composed of the optimal solution of each swarm. The pseudocode of CPSO-S_K is shown in Algorithm 1.

4. The Proposed Method and Analysis

4.1. Stratification Strategy. The initial population must ensure the diversity of the classes of samples. Specifically, we select the initial prototypes from the original dataset using the stratification strategy, which is extracting the prototypes randomly in a certain proportion from different layers of the original dataset. The stratified ratio can be adjusted flexibly.

4.2. Feature Weighting Adjustment. In practical problems, the importance of different features is often different when measuring the similarity between samples. The solution is to give each feature a different weight to represent the importance of the feature. Formula (1) can be improved as

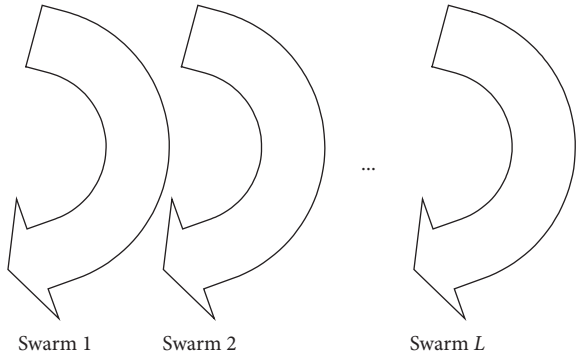


Figure 2: The iterative process of the CCHPSO.

Table 1: Particle representation.

Instance mask						Feature weight			
0	1	—	0	0	1	0.79	0.8	—	0.21

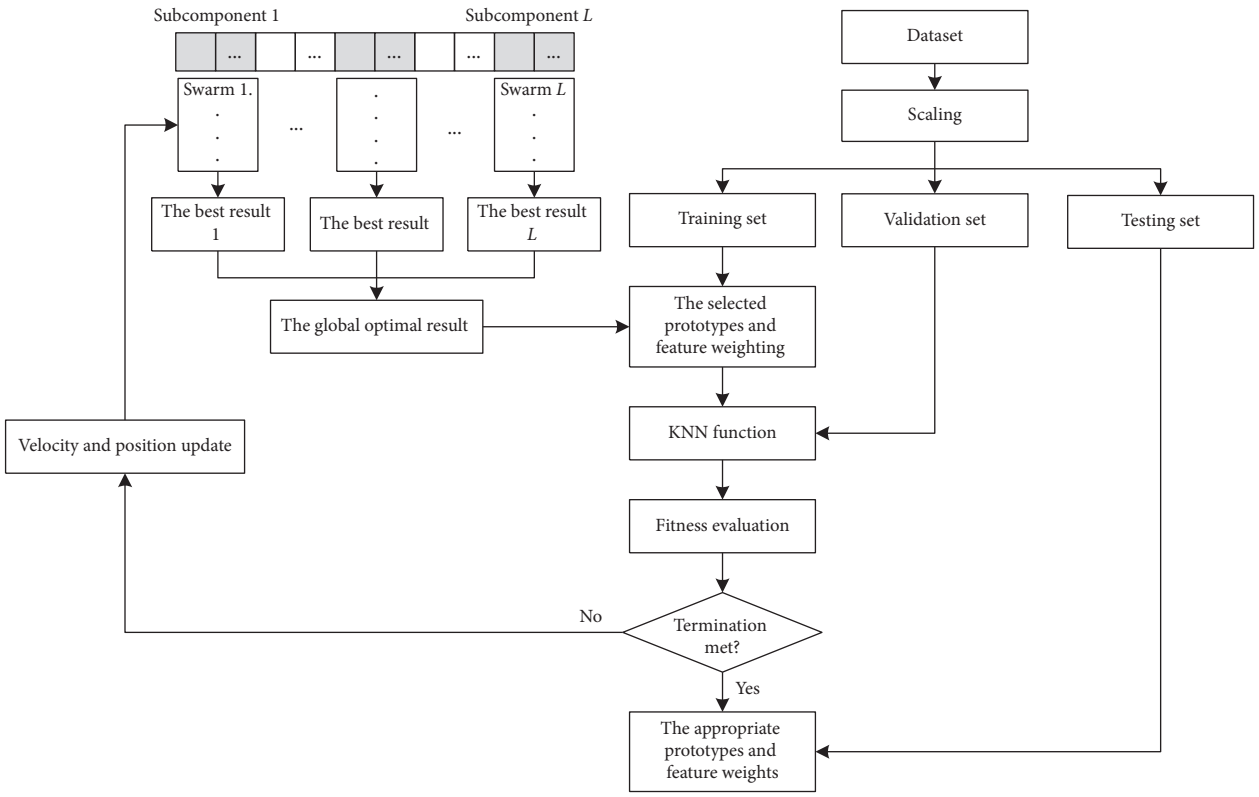


Figure 3: Flow diagram of the proposed CCHPSO-based prototype selection and feature weighting adjustment for KNN in intrusion detection.

$$\mathrm{Fd}\left(x_i, x_j\right) = \sqrt{\sum_{r=1}^{D} \mathrm{Fw}_r \left(x_{ir} - x_{jr}\right)^2}, \tag{5}$$

where $\mathrm{Fd}(x_i, x_j)$ denotes the Euclidean distance between x_i and x_j that takes into account the feature weighting and Fw_r represents the feature weighting of the r-th feature.

4.3. Block Diagram of the CCHPSO-KNN. This section describes the proposed method for the intrusion detection system. The overall process of the proposed model is illustrated in Figure 1. As the initial dataset is huge, the training dataset will be obtained by the stratification strategy, i.e., ensuring that each class has some prototypes. There may be redundancy or noise points in the data, and CCHPSO is used to make prototype selection and feature weighting

Input: training, validation, and testing datasets with labels, KNN as the main classifier, CCHPSO algorithm
Output: testing accuracy (acc), DR, FPR, and confusion matrix.
(1) Training:
(2) Obtain the training, validation, and testing datasets by the stratification strategy
(3) **repeat**
(4) **for** each swarm
(5) **for** each particle
(6) fitness = KNN (pop, train scale, train label, validation scale, and validation label);
(7) update the local and global Sol;
(8) **end for**
(9) Perform position and velocity updates using (2), (3), or (4)
(10) **end for**
(11) **until** termination is met;
(12) Obtain the appropriate prototypes and feature weights according to the global optimal Sol.
(13) Testing:
(14) [testing accuracy, confusion matrix] = KNN (Sol, prototype data, prototype label, test scale, test label);

Algorithm 2: The proposed method.

Table 2: Features of the KDD99 and NSL.

Feature representation	Feature name
F_1	Duration
F_2	protocol_type
F_3	Service
F_4	Flag
F_5	src_bytes
F_6	dst_bytes
F_7	Land
F_8	wrong_fragment
F_9	Urgent
F_{10}	Hot
F_{11}	num_failed_logins
F_{12}	logged_in
F_{13}	num_compromised
F_{14}	root_shell
F_{15}	su_attempted
F_{16}	num_root
F_{17}	num_file_creations
F_{18}	num_shells
F_{19}	num_access_files
F_{20}	num_outbound_cmds
F_{21}	is_hot_login
F_{22}	is_guest_login
F_{23}	Count
F_{24}	srv_count
F_{25}	serror_rate
F_{26}	srv_serror_rate
F_{27}	rerror_rate
F_{28}	srv_rerror_rate
F_{29}	same_srv_rate
F_{30}	diff_srv_rate
F_{31}	srv_diff_host_rate
F_{32}	dst_host_count
F_{33}	dst_host_srv_count
F_{34}	dst_host_same_srv_rate
F_{35}	dst_host_diff_srv_rate
F_{36}	dst_host_same_src_port_rate
F_{37}	dst_host_srv_diff_host_rate
F_{38}	dst_host_serror_rate
F_{39}	dst_host_srv_serror_rate
F_{40}	dst_host_rerror_rate
F_{41}	dst_host_srv_rerror_rate

Table 3: Confusion matrix.

	Predicted attack	Predict normal
Attack	(TP)	(FN)
Normal	(FP)	(TN)

adjustment. Finally, the KNN model will be used to classify the test dataset based on the generated prototypes and feature weights. The dataset we used will be divided into three parts: the training dataset, validation dataset, and testing dataset. The training dataset will be used to produce the prototypes and the feature weights, the validation dataset is employed to validate the feasibility of the selected prototypes and feature weights during the training process, and the generated prototype and feature weights will be used to test the test dataset in the last step.

In the first stage, CCHPSO is used to select the instance subset and feature weight. The D-dimensional object vector is decomposed into L subcomponents illustrated in Figure 2; i.e., each of the L-subcomponents corresponds to a swarm which has s-dimensions selected from the D-dimensional object vector ($D = L * s$). The arrow in Figure 2 indicates the iterative process of each swarm which will output a best result after it evolves. The iterative process of the cooperatively coevolving algorithm is just like unlocking a suitcase's password lock. The global optimal results can be obtained by combing the results evolved from different subcomponents.

In particular, the particle and the fitness function need to be defined first. A particle is comprised of two parts including the instance mask and feature weight. The structure of a particle is shown in Table 1. The first half of the table is a binary string which represents the instance is selected or not, and the second half of the table denotes the feature weights. Suppose there are n instances and m features, and thus there are a total of $n + m$ bits of the particle.

In this model, the high classification accuracy and the few instances are the criteria to design a fitness function. Thus, the fitness function can be defined as

Table 4: Experimental results with the public datasets when $K = 1$ (%).

Datasets	Methods	Training acc	Testing acc	DR	FPR	*R*rate
Kdd	CCHPSO	98.24	97.07	99.01	2.25	92.01
	No selection	98.61	97.70	99.32	3.5	0
	No weighting	98.40	96.72	98.82	3.5	91.85
Nsl	CCHPSO	95.74	90.86	97.42	10	87.26
	No selection	97.70	95.75	98.14	7.50	0
	No weighting	95.81	90.66	96.78	8.25	86.97

Table 5: Experimental results with the public datasets when $K = 3$ (%).

Datasets	Methods	Training acc	Testing acc	DR	FPR	*R*rate
Kdd	CCHPSO	97.15	96.30	98.61	3.75	89.97
	No selection	98.05	96.72	98.53	4	0
	No weighting	96.66	96.16	98.12	3	89.83
Nsl	CCHPSO	94.07	89.61	96.24	9.25	83.20
	No selection	96.93	94.63	98.24	11.25	0
	No weighting	94.36	90.93	97.38	6.75	85.02

Table 6: Experimental results with the public datasets when $K = 5$ (%).

Datasets	Methods	Training acc	Testing acc	DR	FPR	*R*rate
Kdd	CCHPSO	96.46	95.89	98.51	4.75	88.75
	No selection	97.21	95.96	98.32	4	0
	No weighting	96.05	95.47	98.02	5	89.41
Nsl	CCHPSO	92.53	88.91	96.42	10.5	82.32
	No selection	96.51	94.63	98.23	10	0
	No weighting	93.37	90.00	96.85	8.25	82.65

$$\text{fitness} = \omega_1 \times \text{acc} + \omega_2 \times R\text{rate}, \tag{6}$$

where "acc" denotes the classification accuracy based on the current chosen instances, *R*rate represents the reduction rate, w_1 is the weight for the classification accuracy, and w_2 is the weight for the instance selection evaluation. The flow diagram of the proposed method is shown in Figure 3, and the pseudocodes for the proposed method are shown in Algorithm 2.

5. Experiments

5.1. Dataset Used for Experiments. The KDD99 [32] and NSL-KDD [33] were used to demonstrate the generalization ability of the proposed method. Over the years, KDD99 is still recognized as the standard dataset in the field of IDS. Each network connection in the KDD99 and NSL-KDD dataset is described by 41 features shown in Table 2. The types of samples are divided into five categories, including Normal, Probe, DoS, U2R, and R2L. The NSL is more demanding on the IDS method in which duplicate records were removed so that the sample types can reach a balance.

5.2. Evaluation. The accuracy (Acc), detection rate (DR), and false-positive rate (FPR) are used to assess the performance of the intrusion detection method. The above indexes can be obtained by the confusion matrix shown in Table 3, and Acc, DR, and FPR can be expressed as follows.where TP represents the number of attacks correctly recognized, FP represents the number of normal records predicted as attack, FN denotes the number of attacks recognized as normal, and TN represents the number of normals correctly classified.

$$\begin{aligned} \text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{DR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}, \end{aligned} \tag{7}$$

5.3. Experimental Results. The population size for each swarm and the number of iterations are set to be 50 and 200, respectively. Under the dual factors of acc and *R*rate, CCHPSO can iteratively select prototypes and adjust the feature weighting. w_1 and w_2 in the fitness function are set to be 0.9 and 0.1, respectively. All experiments were run on the Matlab R2012a platform equipped with a 2.4 GHz CPU and 32 GB of RAM.

The selection of parameter *K* directly affects the output of KNN. Since the individual is comprised of two parts including the instance mask and feature weight, the experimental results of without considering the prototype selection (no selection)

and without considering the feature weights adjustment (no weighting) under different K values are analyzed. Tables 4–6 show the experimental results of CCHPSO, the no selection method and the no weighting method, respectively, on KDD and NSL datasets when $K = 1$, 3, and 5. The evaluation criteria include the training acc, testing acc, DR, FPR, and Rrate. All results are averaged by ten experiments.

Since the NSL is more demanding on the algorithms, the experimental results based on KDD are generally better than those based on NSL. The accuracy of CCHPSO is 97.07% and 90.86%, respectively, and the false alarm rate is 2.25% and 10% when $K = 1$. It can be concluded that the experimental results are more stable and effective when $K = 1$.

Overall, the no selection method performs best. Because the feature weight is optimized and there is no prototype selection, the prototype data remains intact. It also shows that the reduction rate of the prototype selection using CCHPSO is very high, ranging from 82.32% to 92.01%. From Tables 4–6, we can see that when the data are reduced by about 90%, and the accuracy and other indicators are not greatly affected.

6. Conclusions

The machine learning algorithms are seriously challenged by large datasets, and KNN is one of the most relevant algorithms in machine learning. In this paper, a neighbor prototype selection method based on CCHPSO has been proposed for intrusion detection. The KNN is chosen as the base classier, and the PSO, which can be implemented easily and has few parameters to tune, is used to select prototypes and adjust feature weighting. Moreover, to deal with large-scale optimization problems, a cooperatively coevolving method based on hybrid standard PSO and binary PSO, which employs the divide-and-conquer strategy, is employed. The training samples are generated via the stratification strategy which can ensure the diversity of the classes of samples. Finally, the KNN model is used to classify the test dataset based on the generated prototypes and feature weights. The experiments were conducted on two public datasets to evaluate the effectiveness of the CCHPSO and no selection and no weighting methods. The experimental results show that the reduction rate of the prototype selection using CCHPSO is very high, reaching 92.01%. It can also be concluded that when the data are reduced by about 90%, the accuracy and other indicators are not greatly affected. To advance the execution efficiency, the next step is to improve the model based on GPU parallel computing.

Data Availability

The data supporting this article are from previously reported studies and datasets, which have been cited.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFB0802803) and the National Natural Science Foundation of China (61602052).

References

[1] J. M. Ehrenfeld, "WannaCry, cybersecurity and health information technology: a time to act," *Journal of Medical Systems*, vol. 41, no. 7, p. 104, 2017.

[2] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public cloud systems," *IEEE System Journal*, vol. 13, no. 2, pp. 1478–1486, 2019.

[3] M. Zhang, Y. Yao, Y. Jiang, B. Li, and C. Tang, "Accountable mobile e-commerce scheme in intelligent cloud system transactions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1889–1899, 2018.

[4] A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[5] W. Meng, W. Li, and L.-F. Kwok, "EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism," *Computers & Security*, vol. 43, no. 6, pp. 189–204, 2014.

[6] W. Meng, W. Li, and L. F. Kwok, "Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection," *Security & Communication Networks*, vol. 8, no. 18, pp. 3883–3895, 2015.

[7] C.-F. Tsai and C.-Y. Lin, "A triangle area based nearest neighbors approach to intrusion detection," *Pattern Recognition*, vol. 43, no. 1, pp. 222–229, 2010.

[8] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, no. 1, pp. 13–21, 2015.

[9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Portugal, January 2018.

[10] P. Kuttranont, K. Boonprakob, C. Phaudphut et al., "Parallel KNN and neighborhood classification implementations on GPU for network intrusion detection," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2, pp. 29–33, 2017.

[11] T. Chen, X. Zhang, S. Jin, and O. Kim, "Efficient classification using parallel and scalable compressed model and its application on intrusion detection," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5972–5983, 2014.

[12] A.-F. Farhan, Z. M. Dahalin, and J. Shaidah, "Distributed and cooperative hierarchical intrusion detection on MANETs," *International Journal of Computer Applications*, vol. 12, no. 5, pp. 32–40, 2010.

[13] U. Garain, "Prototype reduction using an artificial immune model," *Pattern Analysis & Applications*, vol. 11, no. 3-4, pp. 353–363, 2008.

[14] S. Garcia, J. Derrac, J. R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[15] I. Triguero, J. Derrac, S. Garcia, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest

neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 86–100, 2012.

[16] N. Verbiest, S. Vluymans, C. Cornelis, N. García-Pedrajas, and Y. Saeys, "Improving nearest neighbor classification using ensembles of evolutionary generated prototype subsets," *Applied Soft Computing*, vol. 44, pp. 75–88, 2016.

[17] W. Hu and Y. Tan, "Prototype generation using multi-objective particle swarm optimization for nearest neighbor classification," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2719–2731, 2016.

[18] L. Nanni and A. Lumini, "Particle swarm optimization for prototype reduction," *Neurocomputing*, vol. 72, no. 4–6, pp. 1092–1097, 2009.

[19] I. Triguero, S. García, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44, no. 4, pp. 901–916, 2011.

[20] M. Rezaei and H. Nezamabadi-Pour, "Using gravitational search algorithm in prototype generation for nearest neighbor classification," *Neurocomputing*, vol. 157, pp. 256–263, 2015.

[21] J. Derrac, I. Triguero, S. Garcia, and F. Herrera, "Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 5, pp. 1383–1397, 2012.

[22] J. Pérez-Rodríguez, A. G. Arroyo-Peña, and N. García-Pedrajas, "Simultaneous instance and feature selection and weighting using evolutionary computation: proposal and study," *Applied Soft Computing*, vol. 37, pp. 416–443, 2015.

[23] H. J. Escalante, M. Marin-Castro, A. Morales-Reyes et al., "MOPG: a multi-objective evolutionary algorithm for prototype generation," *Pattern Analysis and Applications*, vol. 20, no. 1, pp. 33–47, 2017.

[24] A. A. Kardan, A. Kavian, and A. M. Esmaeili, "Simultaneous feature selection and feature weighting with K selection for KNN classification using BBO algorithm," in *Proceedings of the 5th Information and Knowledge Technology*, pp. 349–354, IEEE, Shiraz, Iran, May 2013.

[25] N. García-Pedrajas, A. de Haro-García, and J. Pérez-Rodríguez, "A scalable approach to simultaneous evolutionary instance and feature selection," *Information Sciences*, vol. 228, no. 7, pp. 150–174, 2013.

[26] A. D. Haro-García and N. García-Pedrajas, "A divide-and-conquer recursive approach for scaling up instance selection algorithms," *Journal Data Mining and Knowledge Discovery*, vol. 18, no. 3, pp. 392–418, 2009.

[27] I. Triguero, D. Peralta, J. Bacardit, S. Garcrd, and F. Herrera, "MRPR: a MapReduce solution for prototype reduction in big data classification," *Neurocomputing*, vol. 150, no. 150, pp. 331–345, 2015.

[28] H. J. Escalante, M. Graff, and A. Morales-Reyes, "PGGP: prototype generation via genetic programming," *Applied Soft Computing*, vol. 40, pp. 569–580, 2016.

[29] R. Paredes and E. Vidal, "Learning prototypes and distances: a prototype reduction technique based on nearest neighbor error minimization," *Pattern Recognition*, vol. 39, no. 2, pp. 180–188, 2006.

[30] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pp. 4104–4108, Orlando, FL, USA, October 1997.

[31] F. VandenBergh and A. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[32] Archibe, U. K. KDD Cup 1999 Data, 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[33] Archibe, U. K. NSL Data, 2006, http://nsl.cs.unb.ca/NSL-KDD.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 6293970, 12 pages
https://doi.org/10.1155/2019/6293970

Research Article

An Approach Enabling Various Queries on Encrypted Industrial Data Stream

Tao Wang,[1,2,3] **Bo Yang**,[1,2] **Guoyong Qiu**,[1] **Lina Zhang**,[1,4] **Yong Yu**,[1] **Yanwei Zhou**,[1] **and Juncai Guo**[1]

[1] *School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*
[2] *State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences), Beijing 100093, China*
[3] *Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, Shaanxi 710016, China*
[4] *Department of Computing Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China*

Correspondence should be addressed to Bo Yang; byang@snnu.edu.cn

Received 14 March 2019; Accepted 11 June 2019; Published 3 July 2019

Guest Editor: Lein Harn

Massive data are generated and collected by devices in the industrial Internet of Things. Data sources would encrypt the data and send them to the data center through the gateway. For some supervision purpose, the gateway needs to observe the encrypted data stream and label the suspicious data. Instead of decrypting ciphertext at the gateway, which is not efficient, this paper presents a Φ-searchable functional encryption scheme that supports inner product evaluations on encrypted data. Based on this scheme, an approach enabling various queries on the encrypted industrial data stream is proposed. The adaptive security of our proposed underlying functional encryption scheme can be proven under general subgroup decision assumptions, and our scheme has the smaller public key, the smaller secret key, and the smaller ciphertext size compared to the related schemes. In addition, the experimental results show that our proposed scheme is efficient. Especially for the gateway, querying on the encrypted data only needs less than 20ms, which is practical for industrial data stream auditing scenario.

1. Introduction

1.1. Motivation. While manufacturers, mobile end systems, security cameras, wearable devices, and so forth have been generating highly distributed data from various systems, devices, and applications in industrial Internet of Things (iIoT), more and more data are gathered and intensively exploited by many organizations to extract valuable information either to make marketing decisions, track specific behaviours, or detect threat attacks. Big Data gives a huge opportunity to industries and decisions-makers, but it also represents a big risk for users. Due to data breaches, private information is leaked now and then [1, 2]. It is clear that safeguarding private data to protect manufacturers, sensitive customers, or patients is paramount [3, 4]. But, for massive manufacturers and health and financial organizations, actually implementing the best controls and security is challenging, especially when troves of data originate from multiple sources and are stored across singular or multiple databases and data warehouses.

Using encryption for sensitive information can effectively protect privacy [5]. But, paradoxically, encryption will destroy the usability of data. Especially for real-time industrial application's traffic, how to monitor or audit the encrypted data stream is a key problem. For example, as shown in Figure 1, a card payment gateway would observe the transaction stream, which often includes encrypted data between acquiring banks and issuing banks. The payment gateway needs to audit all encrypted data streams to label some suspicious transactions, say, whose value is over $10000. One solution is to encrypt all transactions under the card company's public key and give the private key to the payment gateway, which can decrypt the transactions stream to do auditing. This solution has two obvious drawbacks. One is being not efficient, because decryption needs to be done for every transaction passing by. Another drawback is that it is

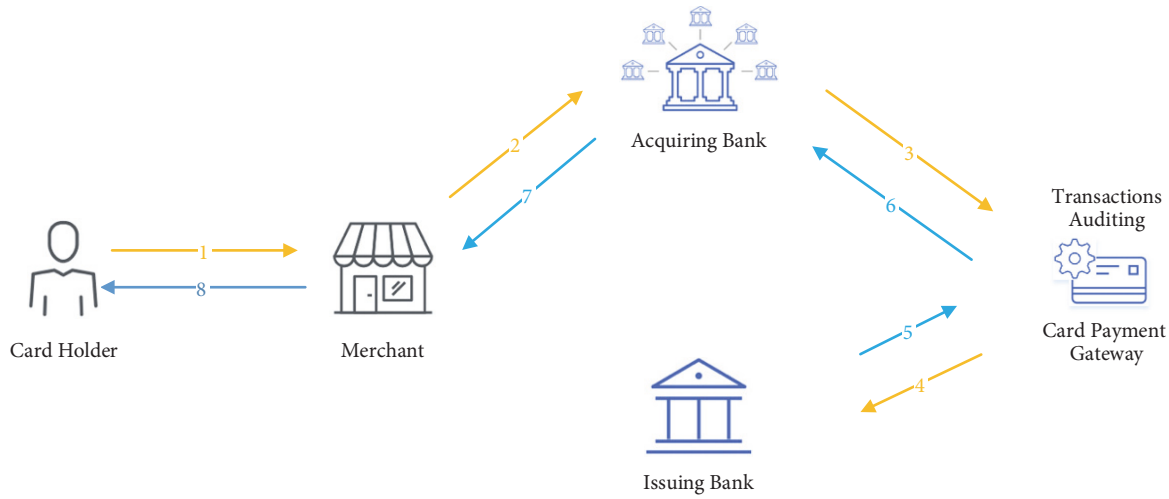


Figure 1: Card payment gateway observes transactions.

bad for both security and privacy concerns, because when the gateway holds the card company's private key, he can see everything he wants.

This work proposes a Φ-searchable functional encryption scheme that supports various types of queries on the encrypted data such as conjunctions, disjunctions, DNFs/CNFs, polynomial equations, and inner products. By utilizing this scheme, we show an approach that can enable the auditing gateways to query or to evaluate the encrypted data stream passing by.

1.2. Related Work. Querying on encrypted data is a long-term interesting open problem in applications with secure and privacy-preserved concerns. Encryption schemes supporting queries on ciphertext are called searchable encryption (SE) schemes, which have two different types of research roadmap. One is searchable symmetric encryption (SSE) [6–12]. The other is public key encryption with keyword search (PEKS) [13–16]. SSE is more efficient but less expressive, and it is hard to achieve privacy for search pattern and access pattern. PEKS is easy to support phrase search and even more complicated evaluations on encrypted data, such as conjunctive, subset, range [14–16], DNF/CNF, polynomial equation, inner product [15, 16], and negation [16]. However, PEKS often has much more computation overheads than SSE due to doing paring computations. There are more recent works for improving security [17, 18], improving functionality [19, 20], and improving performance [21] for the PEKS. Boneh et al. showed that PEKS implies IBE [13]. So, in some sense, an SE scheme is a special form of functional encryption. Brent Waters first publicly used the notion of functional encryption in his talk *Functional Encryption: Beyond Public Key Cryptography*, and Boneh et al. formally defined functional encryption [22]. There was a long-standing open feasibility problem in cryptography: *Does there exist a functional encryption scheme supporting all polynomial-size circuits?* Until 2013, Garg et al. [23] had shown a method to construct functional encryption schemes for polynomial-size circuits based on the indistinguishability obfuscation [24]. Functional encryption has most powerful expressive ability, which can express identity-based encryption (IBE) [25, 26], ABE [27, 28], predicate encryption [14, 15], and inner product encryption [15, 29]. In functional encryption systems, a user who has a decryption key can learn a function of ciphertext. Roughly speaking, in a functional encryption system for functionality $F(\cdot,\cdot)$, an authority holding a master secret key can generate a key $sk_{\mathbf{K}}$ in which the key attribute $\mathbf{K}$ is encoded and that enables the computation of the function $F(\mathbf{K},\cdot)$ on encrypted data which encoded the ciphertext attribute $\mathbf{A}$. More specifically, the user can compute $F(\mathbf{K}, M)$ using $sk_{\mathbf{K}}$ from encryption of plaintext M. We will show the formal definition and security notion of the functional encryption in Section 2.

1.3. Our Approaches and Results. Boneh, Sahai, and Waters have firstly presented formal syntax and put forth a general framework of functional encryption [22]. They also defined two subclasses of functional encryption which are predicate encryption and predicate encryption with public index. In predicate encryption subclass, a functional encryption scheme is defined in terms of a polynomial-time predicate $P : \Sigma_{\mathbf{K}} \times \Sigma_{\mathbf{A}} \longrightarrow \{0,1\}$, where $\Sigma_{\mathbf{K}}$ is key attributes space and $\Sigma_{\mathbf{A}}$ is ciphertext attributes space. Formally, the functional encryption is defined as

$$F\left(\mathbf{K} \in \Sigma_{\mathbf{K}}, (\mathbf{A}, M)\right) := \begin{cases} M & if\ P_{\mathbf{K}}(\mathbf{A}) = 1 \\ \perp & if\ P_{\mathbf{K}}(\mathbf{A}) = 0 \end{cases} \quad (1)$$

Consequently, if $P_{\mathbf{K}}(\mathbf{A}) = 1$, decryption algorithm can recover plaintext and, otherwise, get nothing about the plaintext.

Inspired by the Φ-searchable public key system proposed by Boneh and Waters [9], this paper describes a new functional encryption construction, that is, Φ-searchable functional encryption system providing security against adaptive adversaries and supporting conjunctive, subset,

range, DNF/CNF, polynomial equation, and inner product on encrypted data. We use the predicate encryption subclass to express our functional encryption scheme. We will show the formal definition and security notion of the Φ-searchable functional encryption system in Section 3.

For encoding the key attribute into the secret key and encoding the ciphertext attribute into ciphertext, we follow the inner product encryption (IPE) methodology [15, 29] to realize the predicate $P_{\mathbf{K}}(\mathbf{A})$, which means $P_{\mathbf{K}}(\mathbf{A}) = 1$ if the inner product of key attribute vector $\mathbf{K}$ and ciphertext attribute vector $\mathbf{A}$ is 0 and $P_{\mathbf{K}}(\mathbf{A}) = 0$ otherwise. Formally, for some $\mathbf{K} \in \Sigma_{\mathbf{K}}$ and $\mathbf{A} \in \Sigma_{\mathbf{A}}$, a predicate P over $\Sigma_{\mathbf{K}} \times \Sigma_{\mathbf{A}}$ is defined as

$$P_{\mathbf{K}}(\mathbf{A}) \coloneqq \begin{cases} 1 & if \ \langle \mathbf{K}, \mathbf{A} \rangle = 0 \\ 0 & Otherwise \end{cases} \tag{2}$$

Thanks to "inner-product" style construction, our scheme supports any kind of inner product queries on encrypted data. Clearly, our scheme supports the equality test directly. To achieve this, for the attribute $\mathbf{A}'$ set $\mathbf{A} \coloneqq (-\mathbf{A}', 1)$ and encrypt a message M with $\mathbf{A}$. In order to generate a secret key for the attribute $\mathbf{K}'$, set $\mathbf{K} \coloneqq (1, \mathbf{K}')$. Since $\langle \mathbf{K}, \mathbf{A} \rangle = 0$ if and only if $\mathbf{K}' = \mathbf{A}'$, correctness and security follow. Our scheme also supports the polynomial evaluation after we encode the coefficient of a univariate polynomial into secret keys and encode the univariate into ciphertexts. As a positive result, we can use the polynomial evaluation to achieve supporting conjunctions, disjunctions, CNF, and DNF formulas. We defer the details of applications of our scheme to Section 6.

Our construction relies on general subgroup decision assumptions in composite-order groups which are described in Section 2. We follow the standard Lewko-Waters [30] proof methodology to prove adaptive security of our construction. We propose a Φ-searchable functional encryption system which supports various evaluations on encrypted data including equality, comparison, subset tests, and polynomial evaluations as well as conjunctions, disjunctions, CNF, and DNF formulas. Moreover, compared to the prior constructions that are built for inner product evaluations on encrypted data in composite-order bilinear groups [15, 31], our scheme not only has the adaptive security but also has a smaller public key, smaller secret key, and smaller ciphertext size.

From our proposed searchable encryption scheme, we present an approach enabling various queries on encrypted industrial data for general data flow structure, which includes data sources, gateway, and data center. Our proposed approach makes the gateway easily observe the encrypted data stream passing by sent by the data sources to the data center without decryption. Moreover, if the encrypted data passing by is not matching with some condition, the gateway will learn nothing about the data. From performance evaluation results, the gateway's overhead is less than 20ms which is practical for application in the scenario of querying on the encrypted industrial data stream. We will show our proposed approach in Section 6.1.

2. Preliminaries

2.1. Notations. Given two vectors $\mathbf{U} = (u_1, u_2, ..., u_d) \in \mathbb{Z}_N^d$ and $\mathbf{V} = (v_1, v_2, ..., v_d) \in \mathbb{Z}_N^d$, we use the notation $\langle \mathbf{U}, \mathbf{V} \rangle$ to denote dot product $\mathbf{U}^T\mathbf{V}$. For a group element g, we use $g^{\mathbf{U}}$ to denote a vector $(g^{u_1}, g^{u_2}, ..., g^{u_d})$.

2.2. Syntax of Functional Encryption. We now describe the definition of functional encryption for a functionality $F : \Sigma_{\mathbf{A}} \times \Sigma_{\mathbf{K}} \longrightarrow \{0, 1\}$, where $\Sigma_{\mathbf{A}}$ denotes the ciphertext attributes space and $\Sigma_{\mathbf{K}}$ denotes the key attributes space [22].

Definition 1. For F, a functional encryption scheme consists of four PPT algorithms (Setup, Keygen, Enc, and Dec): for all $\mathbf{A} \in \Sigma_{\mathbf{A}}$ and $\mathbf{K} \in \Sigma_{\mathbf{K}}$, the algorithm Setup$(1^\lambda)$ generates public parameters pp and master secret key mk, the algorithm Keygen$(mk, \mathbf{K})$ outputs secret key for $\mathbf{K}$, the algorithm Enc$(pp, M, \mathbf{A})$ generates ciphertext for a message $M \in \mathscr{M}$, and Dec$(sk_{\mathbf{K}}, c)$ uses $sk_{\mathbf{K}}$ to compute $y = F(\mathbf{K}, M)$ from c.

2.3. Security Notion of Functional Encryption. Before defining the security of functional encryption, we need to describe a restriction for the adversary. Observe that, after the adversary gets the secret keys he wants, he will submit two distinct messages $M_0, M_1 \in \mathscr{M}$. The challenger randomly chooses one to encrypt and sends the ciphertext c to the adversary. Therefore, we need to restrict M_0 and M_1 chosen by the adversary and for all $\mathbf{K}$ that the adversary has $sk_{\mathbf{K}}$, we require that

$$F(\mathbf{K}, M_0) = F(\mathbf{K}, M_1) \tag{3}$$

Clearly, if this restriction is not satisfied, that is, if the adversary has $sk_{\mathbf{K}}$ for some $\mathbf{K}$, he can trivially break the semantic security of the scheme by testing whether Dec$(sk_{\mathbf{K}}, c) = F(\boldsymbol{K}, M_0)$ or not.

For a functional encryption scheme $\mathscr{E}$, $b \in \{0, 1\}$ and, for an adversary $\mathscr{A}$, define an experiment as follows:

(i) *Setup*. Run Setup(1^λ), get (pp, mk), and send pp to $\mathscr{A}$.

(ii) *Query phase1*. $\mathscr{A}$ adaptively makes queries by submitting $\mathbf{K}_i \in \Sigma_{\mathbf{K}}$, where $i = 1, 2, ...$, and receives $sk_{\mathbf{K}_i} \longleftarrow$ Keygen$(mk, \mathbf{K}_i)$.

(iii) *Challenge*. $\mathscr{A}$ outputs two messages $M_0, M_1 \in \mathscr{M}$ satisfying the above restriction and receives Enc(pp, M_b).

(iv) *Query phase2*. $\mathscr{A}$ continues to make queries for some $\mathbf{K}_i$ as query phase1 subject to the restriction and finally outputs a bit.

For b, define

$$\mathrm{Adv}_{\mathscr{A}}^{\mathscr{E}}(\lambda) \coloneqq \left| \Pr\left[b' = b\right] - \frac{1}{2} \right| \tag{4}$$

Definition 2. If, for all PPT $\mathscr{A}$, $\mathrm{Adv}_{\mathscr{A}}^{\mathscr{E}}(\lambda)$ is negligible, we say a functional encryption scheme $\mathscr{E}$ is secure.

2.4. Assumptions over Composite-Order Bilinear Groups

Bilinear Groups of Composite Order. Composite-order bilinear groups were first introduced by Boneh et al. [32] and used by many researchers [15, 33, 34]. Let $\mathcal{G}$ be a group generator that takes a security parameter 1^n as input and outputs $(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups of order $N = p \times q \times r$, where p, q, r are three distinct primes, and $\hat{e} : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ is a function satisfying the following properties:

(i) *Bilinear*: $\forall g_1, g_2 \in \mathbb{G}$, $x, y \in \mathbb{Z}_N$, and $\hat{e}(g_1^x, g_2^y) = \hat{e}(g_1, g_2)^{xy}$.

(ii) *Nondegenerate*: $\exists g \in \mathbb{G}$ such that $\hat{e}(g, g)$ has order N in $\mathbb{G}_T$.

(iii) *Cancellation*: let $\mathbb{G}_p$, $\mathbb{G}_q$, and $\mathbb{G}_r$ be subgroups of $\mathbb{G}$ with order p, q, and r, respectively. For some elements h_1 and h_2 from distinct subgroups, we have

$$\hat{e}(h_1, h_2) = 1 \tag{5}$$

To see this, we note that $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$ and also note that if $g \in \mathbb{G}$ is a generator of $\mathbb{G}$, then g^{pq} generates $\mathbb{G}_r$; g^{pr} generates $\mathbb{G}_q$; g^{qr} generates $\mathbb{G}_p$. Hence, for some elements h_1 and h_2 from distinct subgroups (e.g., $h_1 = h_p \in \mathbb{G}_p$ and $h_2 = h_q \in \mathbb{G}_q$), $h_p = (g^{qr})^{\beta}$ (for some β) and $h_q = (g^{pr})^{\gamma}$ (for some γ). So, we note that $\hat{e}(h_p, h_q) = \hat{e}((g^{qr})^{\beta}, (g^{pr})^{\gamma}) = \hat{e}(g^{\beta}, g^{r\gamma})^{pqr} = 1$.

Cryptographic Assumptions. Our construction relies on the general subgroup decision assumptions in composite-order groups [33]. We now give the following three assumptions.

Assumption 3. Let $\mathcal{G}$ be a group generator as above, and define the following distribution:

$$\begin{aligned}
&(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e}) \longleftarrow_{\$} \mathcal{G}(1^n),\\
&N = p \times q \times r,\\
&g \longleftarrow_{\$} \mathbb{G}_p,\\
&X_3 \longleftarrow_{\$} \mathbb{G}_r,\\
&D = ((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g, X_3),\\
&T_0 \longleftarrow_{\$} \mathbb{G}_{pq},\\
&T_1 \longleftarrow_{\$} \mathbb{G}_p,
\end{aligned} \tag{6}$$

where $\mathbb{G}_{pq}$ is the subgroup of $\mathbb{G}$ with order pq. Define $\mathcal{A}$'s advantage in breaking Assumption 3 as

$$\begin{aligned}
&Adv_{\mathcal{G},\mathcal{A}}^{SD1}(\lambda)\\
&\quad := \left|\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]\right|
\end{aligned} \tag{7}$$

Definition 4. For any ppt algorithm $\mathcal{A}$, if $\mathcal{A}$'s advantage in breaking Assumption 3 and $\mathrm{Adv}_{\mathcal{G},\mathcal{A}}^{\mathrm{SD1}}(\lambda)$ is negligible, we say that $\mathcal{G}$ satisfies Assumption 3.

Assumption 5. Let $\mathcal{G}$ be a group generator as above, and define the following distribution:

$$\begin{aligned}
&(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e}) \longleftarrow_{\$} \mathcal{G}(1^n),\\
&N = p \times q \times r,\\
&g, X_1 \longleftarrow_{\$} \mathbb{G}_p,\\
&X_2, Y_2 \longleftarrow_{\$} \mathbb{G}_q,\\
&X_3 Y_3 \longleftarrow_{\$} \mathbb{G}_r,\\
&D = ((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g, X_1 X_2, X_3, Y_2 Y_3),\\
&T_0 \longleftarrow_{\$} \mathbb{G},\\
&T_1 \longleftarrow_{\$} \mathbb{G}_{pr},
\end{aligned} \tag{8}$$

where $\mathbb{G}_{qr}$ is the subgroup of $\mathbb{G}$ with order qr. Define $\mathcal{A}$'s advantage in breaking Assumption 5 as

$$\begin{aligned}
&Adv_{\mathcal{G},\mathcal{A}}^{SD2}(\lambda)\\
&\quad := \left|\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]\right|
\end{aligned} \tag{9}$$

Definition 6. For any ppt algorithm $\mathcal{A}$, if $\mathcal{A}$'s advantage in breaking Assumption 5 and $\mathrm{Adv}_{\mathcal{G},\mathcal{A}}^{\mathrm{SD2}}(\lambda)$ is negligible, we say that $\mathcal{G}$ satisfies Assumption 5.

Assumption 7. Let $\mathcal{G}$ be a group generator as above, and define the following distribution:

$$\begin{aligned}
&(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e}) \longleftarrow_{\$} \mathcal{G}(1^n),\\
&N = p \times q \times r,\\
&\alpha, s \longleftarrow_{\$} \mathbb{Z}_N\\
&g \longleftarrow_{\$} \mathbb{G}_p,\\
&X_2, Y_2, Z_2 \longleftarrow_{\$} \mathbb{G}_q,\\
&X_3 \longleftarrow_{\$} \mathbb{G}_r,\\
&D = ((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g, g^{\alpha} X_2, X_3, g^{s} Y_2, Z_2),\\
&T_0 = \hat{e}(g, g)^{\alpha s},\\
&T_1 \longleftarrow_{\$} \mathbb{G}_T.
\end{aligned} \tag{10}$$

Define $\mathcal{A}$'s advantage in breaking Assumption 7 as

$$\begin{aligned}
&Adv_{\mathcal{G},\mathcal{A}}^{SD3}(\lambda)\\
&\quad := \left|\Pr[\mathcal{A}(D, T_0) = 1] - \Pr[\mathcal{A}(D, T_1) = 1]\right|
\end{aligned} \tag{11}$$

Definition 8. For any ppt algorithm $\mathcal{A}$, if $\mathcal{A}$'s advantage in breaking Assumption 7 and $\mathrm{Adv}_{\mathcal{G},\mathcal{A}}^{\mathrm{SD3}}(\lambda)$ is negligible, we say that $\mathcal{G}$ satisfies Assumption 7.

2.5. Dual System Encryption.

Brent Waters firstly introduced a methodology to build adaptively secure IBE and HIBE

which is dual system encryption [35], and a lot of work relied on this powerful proof tool [33, 36–38]. In dual system encryption schemes, there are two forms of ciphertext and key: normal and semifunctional. The semifunctional ciphertext and key are only used in the hybrid security proof, while normal ciphertext and key are used in the real system. A normal ciphertext can be decrypted correctly by either normal key or semifunctional key. But semifunctional key cannot decrypt a semifunctional ciphertext, whereas only normal key can. The hybrid security games advance one by one, and the first one is real security game, while in the last one the ciphertext is replaced by encryption of a random message. The most important part of the proof is to show two consecutive games are indistinguishable.

3. Definition

We first show a definition of a Φ-searchable functional encryption system inspired by the Φ-searchable public key system proposed by Boneh and Waters [14]. Then, we show the definition of the security notion.

3.1. Φ-Searchable Functional Encryption System. We use Σ to denote a finite set of binary strings and let Φ be a set of predicates over attributes space Σ. A predicate $P \in \Phi$ is a map: $P : \Sigma \longrightarrow \{0, 1\}$. For two attribute vectors $\mathbf{A}, \mathbf{K} \in \Sigma$, we use the notion $P_{\mathbf{K}}(\mathbf{A}) = 1$ to denote that $\mathbf{A}$ satisfies P which is related to $\mathbf{K}$. We also follow Boneh et al. [14] to use the term *GenToken* to denote the algorithm to generate a search or query token instead of the term *GenKey*, and we use the term *Query* to denote the algorithm to query rather than the term *Decrypt*.

Definition 9. For a predicate $P \in \Phi$, a Φ-searchable functional encryption system comprises four algorithms, $Setup(1^\lambda)$, $Encrypt(PK, (\mathbf{A}, M))$, $GenToken(SK, P_{\mathbf{K}})$, and $Query(TK_{\mathbf{K}}, \mathrm{C})$ such that

(i) $Setup(1^\lambda)$: a probabilistic algorithm that takes as input a security parameter λ and outputs the public parameters PP along with the public key PK and the master secret key SK.

(ii) $Encrypt(PK, (\mathbf{A}, M))$: a probabilistic algorithm that takes as input the public key PK and a plaintext pair $(\mathbf{A}, M)$. We consider $\mathbf{A}$ as the searchable attribute vector of the data M. The algorithm outputs a searchable encryption of $(\mathbf{A}, M)$ under the public key PK.

(iii) $GenToken(SK, P_{\mathbf{K}})$: a probabilistic algorithm that takes the secret key SK and a description of a predicate $P_{\mathbf{K}}$ as input and outputs a search token $TK_{\mathbf{K}}$.

(iv) $Query(TK_{\mathbf{K}}, \mathrm{C})$: a deterministic algorithm that takes a token $TK_{\mathbf{K}}$ and a ciphertext C as input and outputs $F(\mathbf{K}, M)$.

For correctness, we require that, for all λ and all $(PP, PK, SK) \longleftarrow \mathrm{Setup}(1^\lambda)$, all $P_{\mathbf{K}} \in \Phi$, any token $TK_{\mathbf{K}} \longleftarrow GenToken(SK, P_{\mathbf{K}})$, and all $\mathbf{A} \in \Sigma$:

(i) If $P_{\mathbf{K}}(\mathbf{A}) = 1$, then $F(\mathbf{K}, M) = M$.

(ii) If $P_{\mathbf{K}}(\mathbf{A}) = 0$, then $F(\mathbf{K}, M) = \perp$ with all but negligible probability.

3.2. Security Notion. We now show a security notion definition of a Φ-searchable functional encryption system.

Definition 10. A Φ-searchable functional encryption system $\mathscr{E}$ defined as above is adaptive secure if, for all PPT adversaries $\mathscr{A}$, the advantage of $\mathscr{A}$ in the following game is negligible in the security parameters λ:

(i) *Setup*. The challenger runs $Setup(1^\lambda)$ and gives the adversary $\mathscr{A}$ the PP and PK.

(ii) *Query phase 1.* $\mathscr{A}$ outputs descriptions of predicates $P_{\mathbf{K}_1}, P_{\mathbf{K}_2}, \ldots, P_{\mathbf{K}_{\ell_1}} \in \Phi$. The challenger responds with the corresponding tokens:

$$TK_{\mathbf{K}_j} \longleftarrow GenToken\left(SK, P_{\mathbf{K}_j}\right). \tag{12}$$

(iii) *Challenge*. $\mathscr{A}$ outputs two pairs of messages $((\mathbf{A}_0, M_0), (\mathbf{A}_1, M_1)) \in \Sigma \times \mathscr{M}$ subject to the following restrictions:

$$P_{\mathbf{K}_j}(\mathbf{A}_0) = P_{\mathbf{K}_j}(\mathbf{A}_1) \tag{13}$$

for all $P_{\mathbf{K}_j}$ in predicates list queried at query phase 1 and if $M_0 \neq M_1$ then

$$P_{\mathbf{K}_j}(\mathbf{A}_0) = P_{\mathbf{K}_j}(\mathbf{A}_1) = 0 \tag{14}$$

These two restrictions ensure that the tokens given to the adversary do not trivially break the challenge. The first restriction ensures that tokens given to the adversary do not directly distinguish $\mathbf{A}_0$ from $\mathbf{A}_1$. The second restriction ensures that the tokens do not directly distinguish M_0 from M_1.

The challenger randomly chooses $b \in \{0, 1\}$ and gives $C \longleftarrow Encrypt(PK, (\mathbf{A}_b, M_b))$ to $\mathscr{A}$.

(iv) *Query phase 2.* $\mathscr{A}$ continues to output adaptively descriptions of predicates $P_{\mathbf{K}_{\ell_1+1}}, \ldots, P_{\mathbf{K}_\ell} \in \Phi$, subject to the two restrictions (13) and (14). The challenger responds with the corresponding tokens $TK_{\mathbf{K}_j} \longleftarrow GenToken(SK, P_{\mathbf{K}_j})$.

(v) *Guess*. $\mathscr{A}$ outputs a bit b' and wins if $b' = b$.

The adversary $\mathscr{A}$'s advantage in breaking $\mathscr{E}$ is $\mathrm{Adv}_{\mathscr{A}} = |\Pr[b' = b] - 1/2|$.

4. Our Construction

Our construction is based on general subgroup decision assumptions in composite-order groups, that is, Assumptions 3, 5, and 7. Let Φ be a set of predicates over Σ (in our construction, $\Sigma := \mathbb{Z}_N^d$); for a predicate $P \in \Phi$, a Φ-searchable functional encryption scheme for P is defined as follows:

(i) *Setup*(1^λ): this algorithm takes the security parameter λ as input. First, it runs $\mathcal{G}(1^n)$ and gets $(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e})$, where $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r (N = p \times q \times r)$. Then, it computes g_p, g_q, g_r as a generator of $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r$, respectively. In addition, it chooses random α, $x \in_R \mathbb{Z}_N^*$, and $\mathbf{X} \in_R \mathbb{Z}_N^d$. Finally, it outputs public parameters $PP := (\mathbb{G}, g_p, g_r)$ along with the public key:

$$PK := \left(g_p^x, g_p^{\mathbf{X}}, \hat{e}\left(g_p, g_p\right)^\alpha\right) \tag{15}$$

It keeps $SK := (g_p^\alpha, x, \mathbf{X})$ private as the master secret key.

(ii) *Encrypt*$(PK, (\mathbf{A}, M))$: let $\mathbf{A} = (a_1, ..., a_d) \in \Sigma$; this algorithm takes the public key PK and a pair $(\mathbf{A}, M)$ as input and chooses random exponent $s \in_R \mathbb{Z}_N^*$; then it outputs C as the ciphertext, where

$$C := \left\{C_0 := \hat{e}\left(g_p, g_p\right)^{\alpha s} \bullet M,\ C_1 := g_p^{s(x\mathbf{A}+\mathbf{X},1)}\right\} \tag{16}$$

(iii) *GenToken*$(SK, P_{\mathbf{K}})$: let $\mathbf{K} = (k_1, ..., k_d) \in \Sigma$; this algorithm takes as input master secret key SK and a predicate $P_{\mathbf{K}}$, in our case that is $\mathbf{K}$ itself, and chooses random $y \in_R \mathbb{Z}_N^*$ and $\mathbf{W} \in_R \mathbb{Z}_N^{d+1}$. Finally, it outputs a token:

$$TK_{\mathbf{K}} := g_p^{(y\mathbf{K}, \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle)} g_r^{\mathbf{W}} \tag{17}$$

(iv) *Query*$(TK_{\mathbf{K}}, \mathrm{C})$: this algorithm takes a token $TK_{\mathbf{K}}$ for a predicate $P_{\mathbf{K}}$ and a ciphertext C as input; it outputs

$$F(\mathbf{K}, M) := \begin{cases} M := \dfrac{C_0}{\hat{e}\left(C_1, TK_{\mathbf{K}}\right)} & P_{\mathbf{K}}(\mathbf{A}) = 1 \\ \perp & P_{\mathbf{K}}(\mathbf{A}) = 0 \end{cases} \tag{18}$$

where

$$P_{\mathbf{K}}(\mathbf{A}) := \begin{cases} 1 & if\ \ \langle \mathbf{K}, \mathbf{A}\rangle = 0 \\ 0 & Otherwise \end{cases} \tag{19}$$

Correctness. Let C and $TK_{\mathbf{K}}$ be as above. Then

$$\begin{aligned} M &= \frac{C_0}{\hat{e}\left(C_1, TK_{\mathbf{K}}\right)} = \frac{\hat{e}\left(g_p, g_p\right)^{\alpha s} \cdot M}{\hat{e}\left(C_1, TK_{\mathbf{K}}\right)} \\ &= \frac{\hat{e}\left(g_p, g_p\right)^{\alpha s} \cdot M}{\hat{e}\left(g_p{}^{s(x\mathbf{A}+\mathbf{X},1)}, g_p{}^{(y\mathbf{K}, \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle)} g_r{}^{\mathbf{W}}\right)} \\ &= \frac{\hat{e}\left(g_p, g_p\right)^{\alpha s} \cdot M}{\hat{e}\left(g_p{}^{s(x\mathbf{A}+\mathbf{X},1)}, g_p{}^{(y\mathbf{K}, \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle)}\right) \cdot \hat{e}\left(g_p{}^{s(x\mathbf{A}+\mathbf{X},1)}, g_r{}^{\mathbf{W}}\right)} \\ &= \frac{\hat{e}\left(g_p, g_p\right)^{\alpha s} \cdot M}{\hat{e}\left(g_p{}^{s(x\mathbf{A}+\mathbf{X},1)}, g_p{}^{(y\mathbf{K}, \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle)}\right)} \\ &= \frac{\hat{e}\left(g_p, g_p\right)^{\alpha s} \cdot M}{\hat{e}\left(g_p^s, g_p\right)^{(x\mathbf{A}+\mathbf{X},1)^T \bullet (y\mathbf{K}, \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle)}} \end{aligned} \tag{20}$$

where

$$\begin{aligned} &(x\mathbf{A} + \mathbf{X}, 1)^T \bullet \left(y\mathbf{K}, \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle\right) \\ &= (x\mathbf{A} + \mathbf{X}, 1)^T \bullet y\mathbf{K} + \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle \\ &= xy\langle \mathbf{X}, \mathbf{K}\rangle + y\langle \mathbf{X}, \mathbf{K}\rangle + \alpha - y\langle \mathbf{X}, \mathbf{K}\rangle \\ &= xy\langle \mathbf{A}, \mathbf{K}\rangle + \alpha \end{aligned} \tag{21}$$

For all $(\mathbf{A}, \mathbf{K}) \in \mathcal{A} \times \mathcal{K}$ such that $P_{\mathbf{K}}(\mathbf{A}) = 1$, which means $\langle \mathbf{K}, \mathbf{A}\rangle = 0 \bmod N$, the data M will be recovered correctly.

5. Security Proof

To prove the security of our Φ-searchable functional encryption scheme, it depends on the above-mentioned assumptions. Before that, we need to define the semifunctional ciphertext and semifunctional token. These additional structures will only be used in our proof, not in the real system.

(i) *Semifunctional ciphertext*: first, we choose randomly $\mathbf{Z}_c \in_R \mathbb{Z}_N^{d+1}$; then we can use the algorithm *Encrypt*$(PK, (\mathbf{A}, M))$ to construct the normal ciphertext as follows:

$$C' := \left\{C_0' := \hat{e}\left(g_p, g_p\right)^{\alpha s} \cdot M,\ C_1' := g_p^{s(x\mathbf{A}+\mathbf{X},1)}\right\}, \tag{22}$$

and we let the semifunctional ciphertext be

$$\widehat{C} := \left\{\widehat{C_0} := C_0',\ \widehat{C_1} := C_1' \cdot g_q^{\mathbf{Z}_c}\right\} \tag{23}$$

(ii) *Semifunctional token*: we also use the algorithm *GenToken*$(SK, P_{\mathbf{K}})$ to generate the normal token $TK_{\mathbf{K}}'$ and choose a random exponent $\mathbf{Z}_k \in_R \mathbb{Z}_N^{d+1}$. Then we can construct the semifunctional token as follows:

$$\widehat{TK_{\mathbf{K}}} = TK_{\mathbf{K}}' \cdot g_q^{\mathbf{Z}_k} \tag{24}$$

Remark. About query (decryption) capabilities, we observe that a normal ciphertext can be decrypted correctly by either normal key or semifunctional key. But semifunctional key cannot decrypt a semifunctional ciphertext, whereas only normal key can because there is an additional blinding factor of $\hat{e}(g_q, g_q)^{\langle \mathbf{Z}_c, \mathbf{Z}_k\rangle}$. But if $\mathbf{Z}_c$ and $\mathbf{Z}_k$ are orthotropic, the query will still work. We use the term nominally semifunctional token following the definition used by Katz et al. [10] which has components of the subgroup $\mathbb{G}_q$.

Now following the dual system encryption methodology presented by Lewko et al. [30], the proof proceeds with a game sequence starting from Game_{Real}, which is the real security game, followed by a restricted game $\mathrm{Game}_{Restricted}$ which is the same as Game_{Real} except that the adversary cannot query for the tokens for attributes that are equal to the challenge attributes $\mathbf{A}_0, \mathbf{A}_1$. Let ℓ be the number of times of token generation queries that adversary makes. Then we define the following Game_k $(0 \le k \le \ell)$ as follows:

(i) Game_0 is the real game except that the challenge ciphertext is semifunctional.

(ii) Game_k ($0 \le k \le \ell$) is the same as Game_0 except that the first i token generation queries are answered by a semifunctional token, and the last $\ell - i$ token generation queries are answered by a normal token.

Following Game_k, last game is Game_{Final}, which is identical to Game_ℓ, but the challenge ciphertext is not for one of the two messages submitted by the adversary but semifunctional encryption of a random message instead. In the following lemmas, we will prove the indistinguishability between two consecutive games and prove that the adversary $\mathcal{A}$'s view in Game_{Final} is statistically independent of challenge bit b'.

Lemma 11. *If there is an algorithm $\mathcal{A}$ that can distinguish $\text{Game}_{Restricted}$ from Game_{Real} with advantage of $Adv^{\mathcal{A}}_{\text{Game}_{Restricted}} - Adv^{\mathcal{A}}_{\text{Game}_{Real}} = \epsilon$, then we can build an algorithm $\mathfrak{B}$ to break Assumption 3 or Assumption 5 with advantage $\epsilon/2$.*

Proof. If there exists an adversary $\mathcal{A}$ whose advantage is a nonnegligible ϵ, we can find a nontrivial factor of N with nonnegligible probability and break Assumption 5. The proof methodology is similar to the proof of Lemma 1 in [33].

$\mathfrak{B}$ sets up the environment for $\mathcal{A}$ according to Game_{Real}. Suppose that $\mathcal{A}$ produces a ciphertext attribute $\mathbf{A}$ such that it is not equal to the challenge attributes $\mathbf{A}_0^*$ and $\mathbf{A}_1^*$. Since $\mathbf{A} \neq \mathbf{A}_0^*$, there exists at least one pair of components a_i and a_i^* such that $a_i \neq a_i^* \bmod N$, and $a_i - a_i^*$ can be divided by q, where a_i is a component of $\mathbf{A}$ and a_i^* is a component of $\mathbf{A}_0^*$. $\mathfrak{B}$ can compute $d = \gcd(a_i - a_i^*, N)$; set $d' = N/d$. Note that q divides d and $N = dd' = p \cdot q \cdot r$. With probability $\epsilon/2$, one of these two cases must occur; that is, p divides d' or $d = p \cdot q$ and $d' = r$. In the case of p dividing d', $g^{d'}$ is the identity. Then, given g, $\mathfrak{B}$ can test whether $T^{d'}$ is the identity. If not, $T \in \mathbb{G}_{pq}$ holds. Otherwise, $T \in \mathbb{G}_p$. Then $\mathfrak{B}$ breaks Assumption 3. In case of $d = p \cdot q$ and $d' = r$, given g, X_1X_2, X_3, Y_2, Y_3, $\mathfrak{B}$ can verify that $(X_1X_2)^d$ is the identity and determine that $d = p \cdot q$. Then $\mathfrak{B}$ can test whether $\hat{e}((Y_2Y_3)^{d'}, T)$ is the identity. If not, then $T \in \mathbb{G}$ holds. Otherwise, $T \in \mathbb{G}_{pr}$. Then $\mathfrak{B}$ breaks Assumption 5. □

Lemma 12. *If there is an algorithm $\mathcal{A}$ that can distinguish Game_0 from $\text{Game}_{Restricted}$ with advantage of $Adv^{\mathcal{A}}_{\text{Game}_0} - Adv^{\mathcal{A}}_{\text{Game}_{Restricted}} = \epsilon$, then we can build an algorithm $\mathfrak{B}$ to break Assumption 3 with advantage $\epsilon/2$.*

Proof. On input $D = ((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_p, X_3)$ and $T \in \{T_0, T_1\}$, where $T_0 \longleftarrow_{\$} \mathbb{G}_p$ and $T_1 \longleftarrow_{\$} \mathbb{G}_{pq}$, $\mathfrak{B}$ simulates $\mathcal{A}$ as follows.

Setup. Choose random $\alpha, x \in_R \mathbb{Z}_N$, and $\mathbf{X} \in_R \mathbb{Z}_N^d$; set $SK := (g^\alpha, x, \mathbf{X})$ and output

$$PP := \left(\mathbb{G}, g_p, g_r\right),$$
$$PK := \left(g_p^x, g_p^{\mathbf{X}}, \hat{e}\left(g_p, g_p\right)^\alpha\right) \tag{25}$$

Token Queries. Each time $\mathfrak{B}$ is asked to provide a token for a predicate $P_{\mathbf{K}_j}$, it chooses random $y_j' \in_R \mathbb{Z}_N^*$ and $\mathbf{W}_j' \in_R \mathbb{Z}_N^{d+1}$ and outputs a token:

$$TK_{\mathbf{K}_j} := g_p^{(y_j'\mathbf{K}_j, \alpha - y_j'\langle \mathbf{X}, \mathbf{K}_j\rangle)} X_3^{\mathbf{W}_j'} \tag{26}$$

Challenge Ciphertext. After receiving two pair of messages $(\mathbf{A}_0, M_0), (\mathbf{A}_1, M_1)$, $\mathfrak{B}$ chooses random $\beta \in_R \{0, 1\}$ and then forms the ciphertext:

$$C_0 := \hat{e}\left(g_p, T\right)^\alpha \cdot M_\beta,$$
$$C_1 := T^{(x\mathbf{A}_\beta + \mathbf{X}, 1)} \tag{27}$$

It sets the $\mathbb{G}_p$ part of T equal g_p^s to implicitly. The correctness of decryption follows clearly. Observe that if $T = T_0 \longleftarrow_{\$} \mathbb{G}_p$, this is a normal ciphertext and we are in $\text{Game}_{Restricted}$. If $T = T_1 \longleftarrow_{\$} \mathbb{G}_{pq}$, this is a semifunctional ciphertext; then we are in Game_0. □

Lemma 13. *If there is an algorithm $\mathcal{A}$ that can distinguish Game_k from Game_{k-1} with advantage of $Adv^{\mathcal{A}}_{\text{Game}_k} - Adv^{\mathcal{A}}_{\text{Game}_{k-1}} = \epsilon$, then we can build an algorithm $\mathfrak{B}$ to break Assumption 5 with advantage $\epsilon/2$.*

Proof. On input $D = ((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_p, X_1X_2, X_3, Y_2Y_3)$ and $T \in \{T_0, T_1\}$, where $T_0 \longleftarrow_{\$} \mathbb{G}$ and $T_1 \longleftarrow_{\$} \mathbb{G}_{pr}$, $\mathfrak{B}$ simulates $\mathcal{A}$ as follows.

Setup. Choose random $\alpha, x \in_R \mathbb{Z}_N$, and $\mathbf{X} \in_R \mathbb{Z}_N^d$; set $SK := (g_p^\alpha, x, \mathbf{X})$ and output

$$PP := \left(\mathbb{G}, g_p, g_r\right),$$
$$PK := \left(g_p^x, g_p^X, \hat{e}\left(g_p, g_p\right)^\alpha\right) \tag{28}$$

Token Queries. When $\mathcal{A}$ requests the ith token for the predicate $P_{\mathbf{K}_i}$, $\mathfrak{B}$ answers the tokens differently according to the following cases:

(i) Case $i < k$: $\mathfrak{B}$ chooses $y_i' \in_R \mathbb{Z}_N^*$ and $\mathbf{W}_i' \in_R \mathbb{Z}_N^{d+1}$ randomly and creates the semifunctional tokens:

$$TK_{\mathbf{K}_i} := g_p^{(y_i'\mathbf{K}_i, \alpha - y_i'\langle \mathbf{X}, \mathbf{K}_i\rangle)} \left(Y_2Y_3\right)^{\mathbf{W}_i'} \tag{29}$$

Observe that this is an identical distribution from semifunctional tokens.

(ii) Case $i > k$: $\mathfrak{B}$ chooses $y_i' \in_R \mathbb{Z}_N^*$ and $\mathbf{W}_i' \in_R \mathbb{Z}_N^{d+1}$ randomly and creates the normal tokens:

$$TK_{\mathbf{K}_i} := g_p^{(y_i'\mathbf{K}_i, \alpha - y_i'\langle \mathbf{X}, \mathbf{K}_i\rangle)} \left(X_3\right)^{\mathbf{W}_i'} \tag{30}$$

(iii) Case $i = k$: $\mathfrak{B}$ chooses $y_k' \in_R \mathbb{Z}_N^*$, $\mathbf{Z}_k \in_R \mathbb{Z}_N^{d+1}$, and $\mathbf{W}_k' \in_R \mathbb{Z}_N^{d+1}$ randomly and creates the normal tokens:

$$TK_{\mathbf{K}_k} := g_p^{(y_k'\mathbf{K}_k, \alpha - y_k'\langle \mathbf{X}, \mathbf{K}_k\rangle)} T^{\mathbf{Z}_k} \left(X_3\right)^{\mathbf{W}_k'} \tag{31}$$

Challenge Ciphertext. After receiving two pair of messages $(\mathbf{A}_0, M_0), (\mathbf{A}_1, M_1)$, $\mathfrak{B}$ chooses $\beta \in_R \{0,1\}$ randomly and sets $\mathbf{Z}_c \in \mathbb{Z}_N^{d+1}$ such that $\langle \mathbf{Z}_c, \mathbf{Z}_k \rangle = 0$ and then forms the ciphertext:

$$C_0 \coloneqq \hat{e}\left(g_p, X_1X_2\right)^{\alpha} \cdot M_\beta,$$
$$C_1 \coloneqq \left(X_1X_2\right)^{(x\mathbf{A}_\beta + \mathbf{X}, 1)} \tag{32}$$

Recall that $X_1X_2 \in \mathbb{G}_{pq}$ and X_1 is the $\mathbb{G}_p$ part in it. This implicitly sets $X_1 \coloneqq g_p^s$. Furthermore, X_2 is the $\mathbb{G}_q$ part in X_1X_2, so there exits some δ such that $X_2 \coloneqq g_q^\delta$ and $\langle \delta(x\mathbf{A}_\beta + \mathbf{X}, 1), \mathbf{Z}_k \rangle = 0$. Then this implicitly sets $\mathbf{Z}_c \coloneqq \delta(x\mathbf{A}_\beta + \mathbf{X}, 1)$. This relationship between $\mathbf{Z}_c$ and $\mathbf{Z}_k$ makes one thing happen; that is, if $\mathfrak{B}$ wants to test whether the token $TK_{\mathbf{K}}$ is semifunctional by creating a semifunctional ciphertext for predicate $P_{\mathbf{K}}$ and trying to decrypt and finish the query, the decryption will succeed no matter what $TK_{\mathbf{K}}$ is due to $\langle \mathbf{Z}_c, \mathbf{Z}_k \rangle = 0$. So, if $T \in \mathbb{G}_{pr}$, then we are in Game_{k-1}. If $T \in \mathbb{G}$, then we are in Game_k. □

Lemma 14. *If there is an algorithm $\mathscr{A}$ that can distinguish Game_{Final} from Game_ℓ with advantage of $Adv^{\mathscr{A}}_{\text{Game}_{Final}} - Adv^{\mathscr{A}}_{\text{Game}_\ell} = \epsilon$, then we can build an algorithm $\mathfrak{B}$ to break Assumption 7 with advantage $\epsilon/2$.*

Proof. On input $\alpha, s \longleftarrow_\$ \mathbb{Z}_N$, $D = ((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_p, g_p^\alpha X_2, X_3, g_p^s Y_2, Z_2)$ and $T \in \{T_0, T_1\}$, where $T_0 = \tilde{e}(g_p, g_p)^{\alpha s}$ and $T_1 \longleftarrow_\$ \mathbb{G}_T$; $\mathfrak{B}$ simulates $\mathscr{A}$ as follows.

Setup. Choose random $\alpha, x \in_R Z_N$ and $\mathbf{X} \in_R Z_N^d$; set $SK \coloneqq (g_p^\alpha, x, \mathbf{X})$ and output

$$PP \coloneqq \left(\mathbb{G}, g_p, g_r\right),$$
$$PK \coloneqq \left(g_p^x, g_p^{\mathbf{X}}, \hat{e}\left(g_p, g_p\right)^\alpha = \hat{e}\left(g_p^\alpha X_2, g_p\right)\right) \tag{33}$$

Token Queries. Each time $\mathfrak{B}$ is asked to provide a token for a predicate $P_{\mathbf{K}_j}$, it randomly chooses $y_j' \in_R \mathbb{Z}_N$, $\mathbf{W}_j' \in_R \mathbb{Z}_N^{d+1}$, and $\mathbf{Z}_k' \in_R \mathbb{Z}_N^{d+1}$ and outputs a semifunctional token:

$$TK_{\mathbf{K}_j} \coloneqq g_p^{(y_j'\mathbf{K}_j, \alpha - y_j'\langle \mathbf{X}, \mathbf{K}_j \rangle)} Z_2^{Z_k'} X_3^{\mathbf{W}_j'} \tag{34}$$

Challenge Ciphertext. After receiving two pairs of messages $(\mathbf{A}_0, M_0), (\mathbf{A}_1, M_1)$, $\mathfrak{B}$ randomly chooses $\beta \in_R \{0,1\}$ and forms the ciphertext:

$$C_0 \coloneqq T \cdot M_\beta,$$
$$C_1 \coloneqq \left(g_p^s Y_2\right)^{(x\mathbf{A}_\beta + \mathbf{X}, 1)} \tag{35}$$

This implicitly sets $\mathbf{Z}_c \coloneqq (x\mathbf{A}_\beta + \mathbf{X}, 1)$ and this has a proper distribution of semifunctional ciphertexts. So, if $T = \hat{e}(g_p, g_p)^{\alpha s}$, then we are in Game_ℓ. If $T = T_1 \longleftarrow_\$ \mathbb{G}_T$, then we are in Game_{Final}. □

Theorem 15. *Under Assumptions 3, 5, and 7, our Φ-searchable functional encryption scheme described in Section 4 is adaptively secure.*

Proof. When Assumptions 3, 5, and 7 hold, we have shown the indistinguishability between two consecutive games by the previous lemmas, which means that the real security game is indistinguishable from the simulated game in Game_{Final}, in which β is theoretically hidden from the adversary. So, the adversary has no advantage in breaking our scheme. □

6. Applications for Various Queries on Encrypted Data

In this section, we show a candidate system structure that enables the auditor to do various queries on the encrypted industrial data stream and discuss how to implement these query types.

6.1. General Structure for Querying on the Encrypted Industrial Data Stream. Based on our proposed Φ-searchable functional encryption scheme, one can easily enable the gateway to query encrypted industrial data stream. Specifically, as shown in Figure 2, data are collected from various sources such as manufactures, security cameras, and GPS chips. Data sources would send data to the data center through the gateway. The data center stores and analyzes these data, and the gateway observes and audits data stream for supervision purpose. For both security and privacy-preserving concerns, data sources encrypt the data stream under the data center's public key PK with the ciphertext attribute $\mathbf{A}$ by invoking the *Encrypt* algorithm. For a predicate P, the gateway sends the key attribute $\mathbf{K}$ to the data center, and the data center invokes the *GenToken* algorithm to delegate a query token $TK_{\mathbf{K}}$ to the gateway instead of sending its secret key SK to the gateway, which is a bad idea. The gateway who has the query token $TK_{\mathbf{K}}$ can make tests on the encrypted data stream by invoking the *Query* algorithm without decryption. If the output is 1, which means $P_{\mathbf{K}}(\mathbf{A}) = 1$, that is, the encrypted data passing by is matched with the conditions, then the gateway can decrypt that data correctly and take further actions, say, label it. If the output is 0, the gateway can learn nothing about that data; this is guaranteed by the security level of our proposed scheme.

6.2. Equality, Comparison, and Subset Queries. Let $\Sigma_{\mathbf{A}} \coloneqq \mathbb{Z}_N^d$, $\mathbf{A}' \coloneqq (a_1, a_2, ..., a_d) \in \Sigma_{\mathbf{A}}$, and a data $M \in \mathscr{M}$; we encrypt a pair $(\mathbf{A}', M)$ using the *Encrypt* algorithm of our scheme. For example, $\mathscr{M}$ is a personal bank transaction, a_1 is the transaction value, a_2 is the card expiration date, and so on. Also, let $\Sigma_{\mathbf{K}} \coloneqq \mathbb{Z}_N^d$ and $\mathbf{K}' \coloneqq (k_1, k_2, ..., k_d) \in \Sigma_{\mathbf{K}}$. We interpret the predicate $P_{\mathbf{K}'}(\mathbf{A}')$ (i.e., if $k_i = a_i$ or not) for equality test. We interpret the predicate $P_{\mathbf{K}'}(\mathbf{A}')$ as a comparison predicate (i.e., if $k_i \geq a_i$ or not) for a comparison test. We interpret $\mathbf{A}'$ as a set and interpret the predicate $P_{\mathbf{K}'}(\mathbf{A}')$ as a subset predicate (i.e., if $k_i \in \mathbf{A}'$ or not) for subset test. Then, to achieve above-mentioned three kinds of tests, for the attribute $\mathbf{A}'$, set $\mathbf{A} \coloneqq (-\mathbf{A}', 1)$ and encrypt a data M using $\mathbf{A}$. To generate a token for the attribute $\mathbf{K}'$,

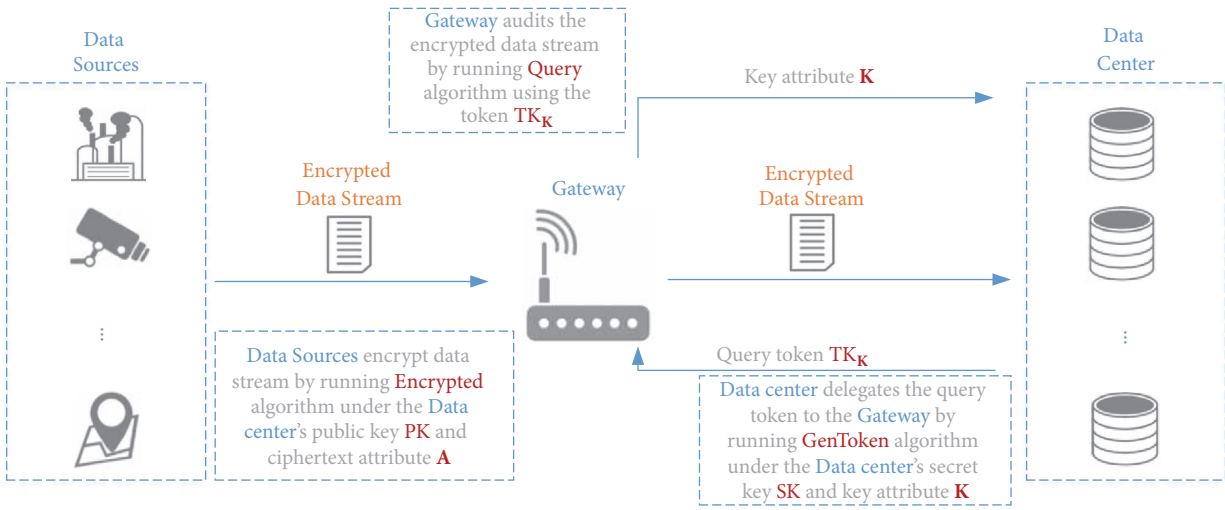


Figure 2: Logical structure for querying on the encrypted industrial data stream.

set $\mathbf{K} := (1, \mathbf{K}')$. Since $\langle \mathbf{K}, \mathbf{A} \rangle = 0$, if and only if $\mathbf{K}' = \mathbf{A}'$, correctness and security follow.

6.3. Polynomial Evaluation. Similar to the predicate encryption scheme presented by Katz et al. [15], we can also support the polynomial evaluation by defining the classes of predicates accordingly. A Φ-searchable functional encryption for polynomials of degree $\leq d(p(x) = k_0 + k_1 x^1 + \cdots + k_d x^d)$ can be defined as follows. Let key attributes space $\Sigma_{\mathbf{K}}^{Poly_{\leq d}} := \mathbb{Z}_p^{d+1}$; we map the polynomial $p(x) = k_0 + k_1 x_1 + \cdots + k_d x^d$ to $\mathbf{K} := (k_0, k_1, \ldots, k_d)$. For ciphertext attribute, each element $w \in \mathbb{Z}_p$ is mapped onto a ciphertext attribute vector $\mathbf{A} := (w^0 \bmod N, w^1 \bmod N, \ldots, w^d \bmod N)$. We also need to define the predicate set $\Phi_{\leq d}^{Poly} := \{P_{\mathbf{K}}^{Poly}(\mathbf{A}) | p \in \mathbb{Z}_N[x], \deg(p) \leq d\}$, where

$$P_{\mathbf{K}}^{Poly}(\mathbf{A}) := \begin{cases} 1 & if\ p(x) = 0 \bmod N \\ 0 & Otherwise \end{cases} \tag{36}$$

for $x \in \mathbb{Z}_N$.

Then, for predicate $P_{\mathbf{K}}^{Ploy}(\mathbf{A}) \in \Phi_{\leq d}^{Ploy}$, correctness and security of our Φ-searchable functional encryption hold, since $p(w) = 0$ whenever $\langle \mathbf{K}, \mathbf{A} \rangle = 0$.

6.4. Conjunctions, Disjunctions, CNF, and DNF Formulas. Based on our Φ-searchable functional encryption for $P_{\mathbf{K}}^{Ploy}(\mathbf{A}) \in \Phi_{\leq d}^{Ploy}$, we can easily support the conjunctions, disjunctions, and their extensions CNF/DNF. We show this ability using an example of conjunctions of equality tests. To do this, for some $\mathbf{K} := (k_0, k_1, ..., k_d)$ and $\mathbf{A} := (a_0, a_1, ..., a_d)$ we define the conjunction predicate as $P_{k_1,k_2}^{\text{AND}}(a_1, a_2)$, where $P_{k_1,k_2}^{\text{AND}}(a_1, a_2) = 1$ if both $k_1 = a_1$ and $k_2 = a_2$. This predicate can be a polynomial as

$$p(x_1, x_2) = r \cdot (x_1 - k_1) + (x_2 - k_2), \tag{37}$$

where $r \longleftarrow_{\$} \mathbb{Z}_N$. If $P_{k_1,k_2}^{\text{AND}}(a_1, a_2) = 1$, then $p(a_1, a_2) = 0$. Otherwise, with all but negligible probability over the choice of r, it will hold that $p(a_1, a_2) \neq 0$.

In a similar fashion, we can define the predicate for disjunction of equality tests. For some $\mathbf{K} := (k_0, k_1, ..., k_d)$ and $\mathbf{A} := (a_0, a_1, ..., a_d)$, we define the disjunction predicate as $P_{k_1,k_2}^{OR}(a_1, a_2)$, where $P_{k_1,k_2}^{OR}(a_1, a_2) = 1$ if either $k_1 = a_1$ or $k_2 = a_2$. This predicate also can be a polynomial as

$$p(x_1, x_2) = (x_1 - k_1) \cdot (x_2 - k_2) \tag{38}$$

If $P_{k_1,k_2}^{OR}(a_1, a_2) = 1$, then $p(a_1, a_2) = 0$; otherwise $p(a_1, a_2) \neq 0$.

We can combine disjunctions, conjunctions, and Boolean variables to handle arbitrary CNF or DNF formulas.

7. Comparison and Evaluation

7.1. Comparison. We compare our construction to prior constructions which are built for inner product evaluations on encrypted data in composite-order bilinear groups [15, 31]. We show the comparison of the basic parameters' performance between these schemes in Table 1.

We use KSW12 to denote the scheme proposed by Katz, Sahai, and Waters [15] and LL18 to denote the scheme proposed by Lee and Lee [31]. As shown in Table 1, our scheme has been proven to be secure against adaptive adversaries, whereas the other two schemes just have selective security. The lengths of the public key are $(3 + 2d)|\mathbb{G}|$ for KSW12 and LL18, whereas $(2 + d)|\mathbb{G}| + |\mathbb{G}_T|$ for our proposal. Obviously, our construction has a smaller public key than others. For the length of the search token (i.e., the private key), our construction has nearly half elements of others. Our construction also gets smaller ciphertext size, $(1+d)|\mathbb{G}|+|\mathbb{G}_T|$, which has just one more group element than LL18 but nearly half elements of KSW12.

Table 1: Comparison of basic parameters∗.

Scheme	Len_{PK}	Len_{TK}	Len_C	SecLev
KSW12 [15]	$(3+2d)\|\mathbb{G}\|$	$(1+2d)\|\mathbb{G}\|$	$(1+2d)\|\mathbb{G}\|$	selectively
LL18 [31]	$(3+2d)\|\mathbb{G}\|$	$2d\|N\|$	$(1+d)\|\mathbb{G}\|$	selectively
Our proposal	$(2+d)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(1+d)\|\mathbb{G}\|$	$(1+d)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	*adaptively*

∗ Let Len_{PK} denote the length of the public key including public parameters, let Len_{TK} denote the length of the search token (in some schemes, i.e., the private key), let Len_C denote the length of the ciphertext, and let SecLev denote the security level. Let $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the length of the element in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively, let $|N|$ denote the length of the element in the field $\mathbb{Z}_N^*$, and let d denote the dimension of the ciphertext attribute and the key attribute.

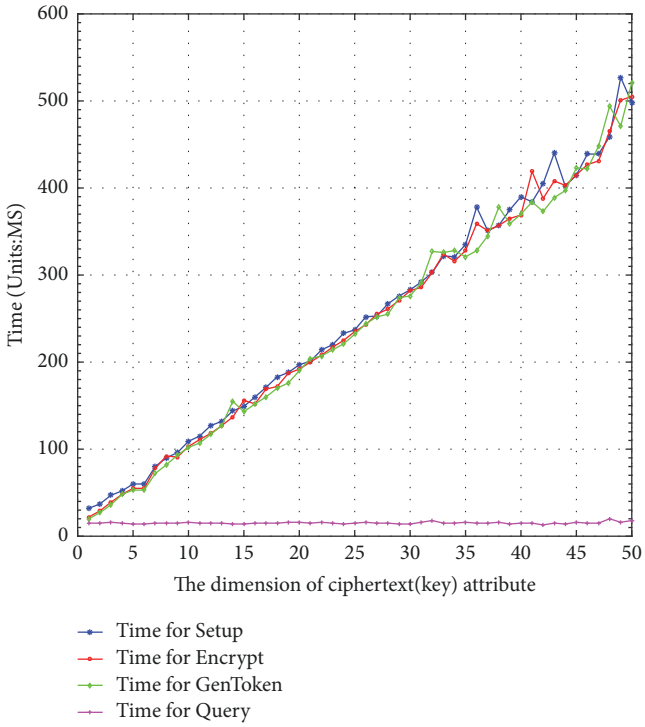


Figure 3: Time cost of the algorithms.

7.2. Performance Evaluation. We implement the algorithms of proposed functional encryption scheme using pairing-based cryptography library pbc-0.5.14 with pbc wrapper-0.8.0 [39] on a PC with 3.3GHz Intel, i5-6600 CPU, and 8GB memory. In our implementation, we made use of parameter *a.param*, one of the standard parameter settings of pbc library. The implementation time overheads are demonstrated as shown in Figure 3. We would like to observe the impact of the dimension of ciphertext (key) attribute vector in terms of the time cost of the algorithms. Obviously, the bigger dimension of ciphertext (key) attribute will make the attributes more expressive, which means the scheme will support more complex predicates. We can see that the time consumptions of the *Setup* algorithm, *Encrypt* algorithm, and *GenToken* algorithm are linearly increasing with the increase from 1 to 50 of the dimension of ciphertext (key) attribute d. The *Setup* algorithm is run by the trusted authority which usually can be executed once and offline. The *Encrypt* algorithm is run by the data source which also can be executed offline. The *GenToken* algorithm is run by the data center that has powerful computing ability. So, the time cost of these three algorithms is considerably acceptable. Fortunately, the *Query* algorithm's time cost is nearly constant (less than 20ms) with the increase of the dimension of ciphertext (key) attribute d. This merit makes the gateway able to effectively test the encrypted data passing by without a significant reduction in processing speed.

8. Conclusions

In this paper, we have put forth an Φ-searchable functional encryption scheme. We have built our scheme on the composite-order bilinear groups and have proven the adaptive security by utilizing dual system encryption proof technology. By using our proposed scheme as the underlying encryption scheme, we present an approach that supports the fact that the gateway effectively audits the encrypted data stream. According to the comparison and performance evaluation results, our proposed encryption scheme has the smaller public key, the smaller query token, and the smaller ciphertext. Moreover, our proposed approach can enable the gateway to effectively test the encrypted data stream, which is practical for industrial data stream auditing scenario.

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 they have no conflicts of interest.

Acknowledgments

This work is supported by National Key R&D Program of China (no. 2017YFB0802000), the National Natural Science Foundation of China (61572303, 61772326, 61802241, and 61802242), National Cryptography Development Fund during the 13th Five-year Plan Period (MMJJ20180217), the Foundation of State Key Laboratory of Information Security (2017-MS-03), the Fundamental Research Funds for the Central Universities (no. GK201903089, no. GK201702004, and no. GK201603084), and the Natural Science Basic Research Plan in Shaanxi Province of China (2019JM-552).

References

[1] Q. Jiang, X. Huang, N. Zhang, K. Zhang, X. Ma, and J. Ma, "Shake to communicate: secure handshake acceleration-based pairing mechanism for wrist worn devices," *IEEE Internet of Things Journal*, 2019.

[2] Q. Jiang, J. Ma, and C. Yang, "Efficient End-To-End Authentication Protocol for Wearable Health Monitoring Systems," *Computers Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[3] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[4] 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 ID e3900, 2019.

[5] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[6] D. X. Song, D. A. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of IEEE Symposium on Security and Privacy. (S&P '2000)*, pp. 44–55, USA, May 2000.

[7] E. J. Goh, "Secure Indexes," IACR Cryptology ePrint Archive: Report 2003/216, 2003.

[8] C. Bösch, A. Peter, B. Leenders et al., "Distributed searchable symmetric encryption," in *Proceedings of the Twelfth Annual International Conference on Privacy, Security and Trust, PST '2014*, pp. 330–337, Canada, July 2014.

[9] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[10] P. Han, C. Liu, B. Fang et al., "Revisiting the practicality of search on encrypted data: from the security brokers perspective , scientific programming," *Scientific Programming*, vol. 2016, 9 pages, 2016.

[11] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proceedings of the ACM conference on Computer and Communications Security, CCS '2012*, pp. 965–976, USA, October 2012.

[12] Z. Wang, Z. Fu, and X. Sun, "Semantic contextual search based on conceptual graphs over encrypted cloud," *Security and Communication Networks*, vol. 2018, pp. 1–10, 2018.

[13] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano et al., "Public key encryption with keyword search," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 506–522, Springer.

[14] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceedings of Fourth IACR Theory of Cryptography Conference, TCC '2007*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 535–554, Springer, 2007.

[15] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," *Journal of Cryptology*, vol. 26, no. 2, pp. 191–224, 2013.

[16] N. Attrapadung and B. Libert, "Functional encryption for inner product achieving constant-sizeciphertexts with adaptive security or support for negation," in *Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography PKC '2010*, vol. 6056 of *Lecture Notes in Computer Science*, pp. 384–402, Springer Berlin Heidelberg, France, 2010.

[17] R. Xie, C. He, D. Xie, C. Gao, and X. Zhang, "A secure ciphertext retrieval scheme against insider kgas for mobile devices in cloud storage," *Security and Communication Networks*, vol. 2018, Article ID 7254305, 7 pages, 2018.

[18] D. Sharma and D. C. Jinwala, "Multiuser searchable encryption with token freshness verification," *Security and Communication Networks*, vol. 2017, 16 pages, 2017.

[19] S. Kamara and T. Moataz, "SQL on structurally-encrypted databases," IACR Cryptology ePrint Archive Report 2016/453, 2016.

[20] D. N. Wu, Q. Q. Gan, and X. M. Wang, "Verifiable public key encryption with keyword search based on homomorphic encryption in multi-user setting," *IEEE Access*, vol. 6, 9 pages, 2018.

[21] G. Asharov, M. Naor, G. Segev et al., "earchable symmetric encryption -optimal locality in linear space via two-dimensional balanced allocations," in *Proceedings of the 48th Annual Symposium on the Theory of Computing, STOC '2016*, pp. 1101–1114, ACM, New York, NY, USA, 2016.

[22] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: definitions and challenges," IACR Cryptology ePrint Archive Report 2010/543, 2010.

[23] S. Garg, C. Gentry, S. Halevi et al., "Candidate indistinguishability obfuscation and functional encryption for all circuits," in *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS '2013*, pp. 40–49, USA, October 2013.

[24] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating eves algorithm and its application in fair electronic transactions in public cloud systems," *IEEE Systems Journal*, 2019.

[25] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, CRYPTO '1984*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, November 2000.

[26] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the Annual International Cryptology Conference, CRYPTO '2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, USA, August 2001.

[27] A. Sahai and B. Waters, "Fuzzy identity-based encryption," Cryptology ePrint Archive Report 2004/086, 2004.

[28] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, USA, November 2006.

[29] T. Okamoto and K. Takashima, "Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption," in *Advances in Cryptology – EUROCRYPT 2012*, vol. 7237 of *Lecture Notes in Computer Science*, pp. 591–608, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[30] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Advances in Cryptology—EUROCRYPT 2012*, vol. 7237 of *Lecture Notes in Computer Science*, pp. 318–335, Springer, Berlin, Heidelberg, Germany, 2012.

[31] K. LEE and D. H. LEE, "Two-Input functional encryption for inner products from bilinear maps," *IEICE Transactions on*

Fundamentals of Electronics, Communications and Computer Sciences, vol. E101.A, no. 6, pp. 915–928, 2018.

[32] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of the Second Theory of Cryptography Conference, TCC '2005*, vol. 3378 of *Lecture Notes in Computer Science*, pp. 325–341, Springer, USA.

[33] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure HIBE with short ciphertexts," in *Theory of Cryptography*, vol. 5978 of *Lecture Notes in Computer Science*, pp. 455–479, Springer, Berlin, Germany, 2010.

[34] D. Boneh, A. Sahai, and B. Waters, "Fully collusion resistant traitor tracing with short ciphertexts and private keys," in *Advances in Cryptology - EUROCRYPT 2006*, vol. 4004 of *Lecture Notes in Computer Science*, pp. 573–592, Springer, Russia, 2006.

[35] B. Waters, "Dual system encryption: realizing fully secure ibe and hibe under simple assumptions," in *Advances in Cryptology—CRYPTO 2009*, vol. 5677 of *Lecture Notes in Computer Science*, pp. 619–636, Springer, 2009.

[36] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Advances in Cryptology—CRYPTO 2010*, T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, pp. 191–208, Springer, Berlin, Germany, 2010.

[37] A. Lewko, Y. Rouselakis, and B. Waters, "Achieving leakage resilience through dual system encryption," in *Theory of Cryptography—TCC 2011*, vol. 6597 of *Lecture Notes in Computer Science*, pp. 70–88, Springer, Berlin, Heidelberg, Germany, 2011.

[38] J. Zhang, J. Chen, A. Ge et al., "Shorter decentralized attribute-based encryption via extended dual system groups," *Security and Communication Networks*, vol. 2017, 19 pages, 2017.

[39] B. Lynn, "The pairing-based cryptography library (0.5.13)," http://crypto.stanford.edu/pbc/.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 8105787, 8 pages
https://doi.org/10.1155/2019/8105787

Research Article

A QR Code Secret Hiding Scheme against Contrast Analysis Attack for the Internet of Things

Qinglan Zhao,[1,2] Shuntong Yang,[1] Dong Zheng,[1,3] and Baodong Qin[1]

[1]*National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*
[2]*Key Laboratory of Applied Mathematics (Putian University), Fujian Province University, Fujian, Putian 351100, China*
[3]*Westone Cryptologic Research Center, Beijing 100070, China*

Correspondence should be addressed to Dong Zheng; zhengdong_xupt@sina.com

Received 14 March 2019; Accepted 9 May 2019; Published 3 July 2019

Guest Editor: Fagen Li

Due to the advantages of larger content and error correction capability, quick response (QR) code is commonly used as a tagging technology for the Internet of Things (IoT) recently. However, the cover message of QR code can be easily decoded by a QR code reader, which causes the security and privacy of the cover message to raise the important issues. In this paper we present a new secret hiding scheme based on QR code. The proposed scheme has low computational complexity and is suitable for low-power devices in IoT systems because of utilizing the error correction property of QR code to hide secret information. The proposed scheme hides the secret information without changing the cover message of QR code and the user can get the cover message by using a general scanner, which contributes to reducing attacker's curiosity. The hidden secret information can be read by a special scanner with the help of the user key. One thing which is better than other known schemes is that the proposed scheme can resist contrast analysis attack. In addition, experimental results show the proposed scheme has feasibility, low computational complexity, and high hiding payload.

1. Introduction

The Internet of Things (IoT) interconnects physical and digital objects that are identifiable and may interact with each other and with users. These objects, each with its own identity, are well beyond only computers and they are our cars, luggage, household applicants, humans, and so on. This was made possible by different tagging technologies like radio frequency identification (RFID) and two-dimensional (2D) barcode which allow physical objects to be identified and refer over the IoT. Due to the less complexity and a cheap solution of 2D barcodes, they have become popular for building an IoT system. Quick response (QR) code, as a 2D barcode with the advantages of larger QR content and error correction capability, is commonly used recently.

QR code can store rich information including text, URL link, and other types of data. They can be used as a data carrier to allow users to access the system more conveniently in an IoT system. However, the cover message of the QR code can be easily read by a barcode reader [1], such as a mobile phone with a camera. This may lead to leaking of privacy. Another important issue is the security of QR code when we use a QR code to communicate secret information.

The traditional method of encrypting secret information into cipher texts makes it impossible for an attacker to obtain secret information, thereby achieving confidentiality [2–4]. This method requires a lot of computation and is sometimes not suitable for the objects of IoT system. It also clearly points out what is important information and easily attracts the attention of the attacker to increase the possibility of being attacked [5]. Information hiding technologies based on QR code have emerged to solve these new problems. The image hiding schemes [6–8] are mainly to convert the secret into a QR code tag and then embed the secret QR code into the image. However, these schemes require complicated image processing operations to recover the hidden QR code. Most watermarking algorithms, which use Discrete Cosine Transform, Discrete Wavelet Transform, and Discrete Fourier Transform algorithms to hide the watermark in the QR code [9–12], have high computational complexity and limited

hidden information caused by the length and width of QR code. Therefore, due to the high computational complexity of these methods, they are not suitable for low-power mobile devices.

To meet the demands of applications of QR code to low-power mobile devices in IoT systems, some schemes have been presented which used the code characteristic of the QR code to hide secret [13–17]. Chiang et al. proposed a scheme [14] to hide the secret information being confused by the pseudo-random binary stream generated by the user key. To increase the hiding payload, [16] proposed a data hiding method which is an extended version of [13]. However, these methods can not resist the contrast analysis attack. Under such an attack scenario, the attacker can contrast the codewords of QR codes which have different cover message and hide the same secret information with the same user key. These methods insert some data related to the secret information into the original codewords and make positions of secret message unchanged when the secret and key do not change. The attacker can get the data which are the same part of these codewords. Even the attacker can not recover the secret from the data they got without the key, they can create a new QR code with embedding these data which hide secret information. When these secret schemes are used for copyright protection, by this method the attacker can forge copyright information containing the legal copyright information.

In order to resist the contrast analysis attack, we design a new QR code secret hiding scheme. The proposed scheme makes the changed codewords of original QR code related to the cover message using the simple XOR operation. Compared with original QR code, QR codes have different changed codewords if they have the different cover message and hide the same secret information with the same key. So the attacker can not find the same data related to the secret information and key through the contrast analysis attack. In addition, the proposed scheme utilizes the biggest error correction ability of QR code to resist brute force attack. With higher security than the known schemes, the proposed scheme has the low computational complexity and high hiding payload.

The paper is organized as follows. Section 2 introduces QR code technique. The proposed secret hiding scheme is described in Section 3. The simulation, performance comparisons, and security analysis are discussed in Section 4. Finally, Section 5 concludes the paper.

2. The Technology of QR Code

QR code is one of the most popular 2D barcodes [18]. It consists of white and black square modules which are equal to the binary values 0 and 1. Figure 1 depicts an instance of QR code symbol. The number of modules increases with QR code version. There are 40 QR code standard versions among which Version 1 has the smallest 21 × 21 modules and Version 40 has the largest 177 × 177 modules. The data payload becomes larger as the version evolves. There is 208 data modules in Version 1 and 29648 data modules in Version 40. The message bit stream shall be divided into codewords. All codewords are 8 bits in length.

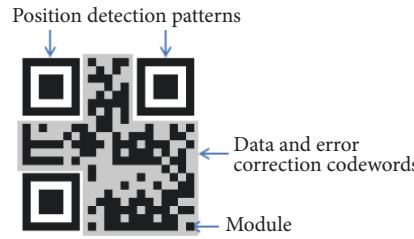


Figure 1: Basic structure of a QR barcode.

Table 1: Error correction levels.

Error Correction Level	Recovery Capacity %(approx.)
L	7
M	15
Q	25
H	30

To achieve the recovery ability, the error correction algorithm has been used in QR code to generate a series of error correction codewords which are added to the data codeword sequence. The error up to 30% can be corrected. Each version has four error correction levels L, M, Q, and H as shown in Table 1. Depending on the version and error correction level, the data codeword sequence is subdivided into one or more blocks, to each of which the error correction algorithm shall be applied separately to the data codeword. To show this, the error correction characteristics of QR code of Version 1, 20, and 40 are listed in Table 2, where c is the total number of codewords, k is the number of data codewords, and r is the number of error correction capacity. For Version 20 with error correction level L, as an example, 1085 codewords are divided into 8 blocks in which 3 blocks apply error correction codewords (135,107,14) and 5 blocks apply error correction codewords (136,108,14).

The process to construct a QR code is structured into seven steps.

Step 1 (data analysis). The input data stream is analyzed to identify the variety of different characters to be encoded and the version and error correction level are selected.

Step 2 (data encodation). Data characters are converted to a bit stream which is split into 8-bit codewords.

Step 3 (error correction coding). The codeword sequence is divided into the required number of blocks and the error correction codewords are calculated for each block with being appended to the end of the data codeword sequence.

Step 4 (structure final messages). The final sequence is assembled by taking data and error correction codewords from each block in turn.

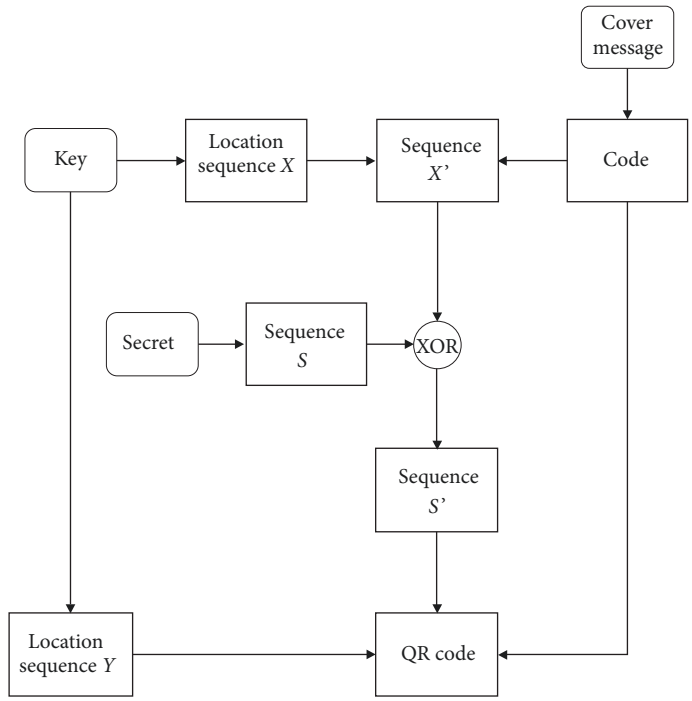


Figure 2: Secret hiding procedure.

Step 5 (module placements in matrix). The codeword modules are placed in the matrix together with other patterns.

Step 6 (masking). To optimize the dark/light module balance and minimize the occurrence of undesirable patterns, the masking pattern is applied.

Step 7 (format and version information). The format and version information are created.

3. The Proposed Scheme

The proposed scheme utilizes the code characteristic of QR code to hide secret information. It includes two procedures, secret hiding procedure and secret extraction procedure, whose flowcharts are shown in Figures 2 and 3. Secret hiding procedure is integrated with the QR code generation for a cover message and secret extraction procedure is integrated with the QR code scanning, as we will describe in detail.

3.1. Secret Hiding Procedure. For the secret information to be hidden, its code MC can be given according to the coding principle of the QR code. MC is expanded to a longest sequence S whose length is related to error correction of the QR code version. Suppose r be the number of error correction capacity. The length of S is $8r$. In order to resist QR contrast attack, S will be confused by a sequence related to the cover message. The following are details of secret hiding procedure.

Step 1. Perform the normal QR code encoding procedure for the secrete information and cover message until their codewords are generated.

Step 2. Use user's key ks as input to generate a location sequence $Y = [Y_0, Y_1, \ldots, Y_{8r-1}]$, where $Y_i = (m, n)$, such that $m \in C'$ with C' being a subset of $\{0, 1, \ldots, c-1\}$ and $|C'| = r$, $0 \le n \le 7$ for all $0 \le i \le 8r-1$.

Step 3. Use user's key ks as input to generate a sequence $X = [X_0, X_1, \ldots, X_{8r-1}]$ with $X_i = (t, s)$, $0 \le t \le c-1, 0 \le s \le 7$ such that the tth codeword is one of data codewords for all $0 \le i \le 8r-1$.

Step 4. Let l be the length of the secret codewords MC and the length of the code of l be q. Sequence $0 \ldots 0$ with length $l = r * 8 - q$ is added to code MC, and then the code of l is added at the end. The resulting sequence is $S = [S_0, S_1, \ldots, S_{8r-1}]$.

Step 5. According to X, find the data X'_i of the block on the position $X_i = (t, s)$, that is, the sth data of the tth codeword, for all $0 \le i < 8r-1$. Then get a sequence $X' = [X'_0, X'_1, \ldots, X'_{8r-1}]$.

Step 6. Generate a sequence $S' = [S'_0, S'_1, \ldots, S'_{8r-1}]$ with $S'_i = S_i \oplus X'_i$ for $0 \le i \le 8r-1$.

Step 7. Embed the sequence S' into the cover message codewords according to the position sequence Y. For $Y_i = (m, n)$ with $0 \le i \le 8r-1$, look for the nth bit of the mth block in the cover message codewords and replace it with S'_i.

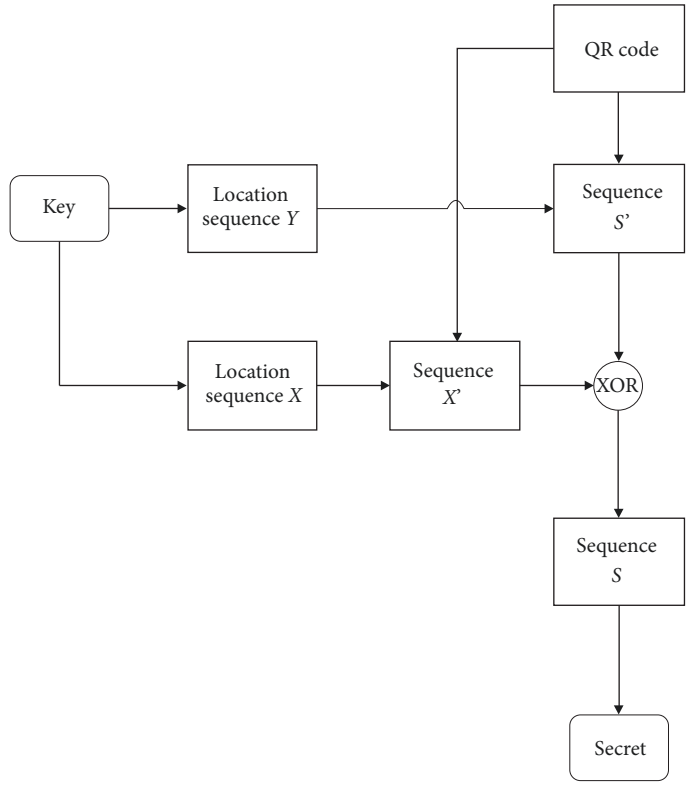


Figure 3: Secret extraction procedure.

Step 8. Continue the process of encoding the QR code with hidden secret information.

3.2. Secret Extraction Procedure. To extract the secret information, we design the special scanner in which the secret extraction is integrated with scanning process. The common scanner only can read the cover message by scanning the QR code. Using the special scanner, the authorized user can get the cover message and the secret information with inputting the key *ks*. Secret extraction procedure includes the following steps.

Step 1. Scan QR code to get data block information before error correction.

Step 2. Use user's key *ks* as input to generate a location sequence Y.

Step 3. According to the location sequence Y, for any $Y_i = (m, n)$ with $0 \le i \le 8r - 1$, find the nth bit data S'_i of the mth codeword in the QR code and get the sequence $S' = [S'_0, S'_1, \ldots, S'_{8r-1}]$.

Step 4. Finish error correction and use user's key as input to generate a sequence X.

Step 5. According to X, find the data X'_i on the position $X_i = (t, s)$ of the block and get the sequence $X' = [X'_0, X'_1, \ldots, X'_{8r-1}]$ corresponding with $X = [X_0, X_1 \ldots, X_{8r-1}]$, $0 \le i \le 8r - 1$.

Step 6. Generate sequence $S = [S_0, S_1, \ldots, S_{8r-1}]$ with $S_i = S'_i \oplus X'_i$ for $0 \le i \le 8r - 1$.

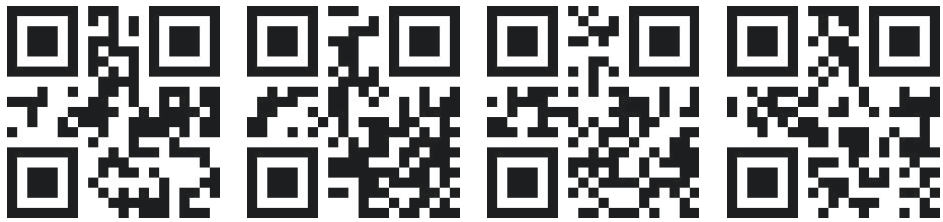
Step 7. Calculate the length l of the secret according to the last qth bits and then get the secret codewords $[S_0, S_1, \ldots, S_{l-1}]$.

4. Simulation Results and Analysis

To assess the feasibility and suitability of the scheme, we implement the proposed secret hiding scheme using Python language which has a powerful operation library of QR code.

4.1. Experimental Results and Practicability. The results of the proposed scheme for the 1-H QR versions are shown in Figure 4. Figure 4(a) is the original QR code image with the cover message "Data". Figure 4(b) is the QR code image which has the same cover message with Figure 4(a) and hides the secret message "Secret" with the user key "google". Figures 4(c) and 4(d) hide the same secret and use the same user key as Figure 4(b), but have the different cover message "Escher" and "Linux".

To show the hiding procedure, we introduce *QRC* which denote the coding function of QR code including step 1 and step 2 of constructing QR code and *QRCS* denote the coding



(a) (b) (c) (d)

Figure 4: The results of 1-H QR code in different cases.

Table 2: Error correction characteristics of Versions 1, 20, and 40.

Version	Total number of codewords	Error correction level	Number of error correction codewords	Number of error correction blocks	Error correction code per block (c,k,r)
1	26	L	7	1	(26,19,2)
		M	10	1	(26,16,4)
		Q	13	1	(26,13,6)
		H	17	1	(26,9,8)
20	1085	L	224	3	(135,107,14)
				5	(136,108,14)
		M	416	3	(67,41,13)
				13	(68,42,13)
		Q	600	15	(54,24,15)
				5	(55,25,15)
		H	700	15	(43,15,14)
				10	(44,16,14)
40	3706	L	750	19	(148,118,15)
				6	(149,119,15)
		M	1372	18	(75,47,14)
				31	(76,48,14)
		Q	2040	34	(54,24,15)
				34	(55,25,15)
		H	2430	20	(45,15,15)
				61	(46,16,15)

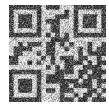
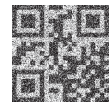
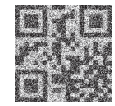
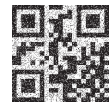
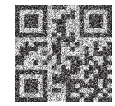
function of QR code with secret hidden. For simplicity, we use a decimal number to represents an 8-bit binary sequence. In Figure 4(a), "Data" is the cover message of the QR code. With using the mask type 7, we get QRC("Data") as follows: QRC(Data) = [64, 68, 70, 23, 70, 16, 236, 17, 236, 97, 53, 255, 172, 71, 43, 105, 94, 82, 129, 51, 201, 118, 131, 139, 97, 120].

In Figure 4(b), the cover message of the QR code is "Data", the user key is "google", the secret message is "Secret", and the mask type is 0. According to secret hiding procedure, we get $S = [115, 101, 99, 114, 101, 116, 0, 6]$. $S' = [155, 75, 232, 240, 129, 46, 118, 94]$. $Y = [(18, 6), (18, 0), (6, 5), (18, 2), (23, 3), (15, 4), (18, 1), (6, 6), (4, 3), (15, 6), (7, 1), (7, 0), (25, 2), (23, 5), (6, 1), (25, 3), (23, 1), (18, 4), (4, 4), (12, 3), (18, 7), (4, 5), (7, 2), (25, 1), (6, 3), (23, 0), (15, 5), (12, 6), (4, 2), (15, 3), (23, 2), (25, 7), (25, 6), (23, 6), (6, 7), (7, 4), (12, 4), (4, 1), (12, 7), (7, 6), (18, 3), (18, 5), (4, 7), (25, 0), (4, 6), (7, 7), (7, 5), (25, 4), (12, 0), (15, 1), (15, 2), (6, 2), (7, 3), (23, 4), (15, 0), (23, 7), (12, 2), (12, 1), (6, 0), (12, 5), (4, 0), (6, 4), (25, 5), (15, 7)]$. $QRCS$(Data)= [64, 68, 70, 23, 139, 16, 122, 7, 236, 97, 53, 255, 70, 71, 43, 230, 94, 82, 107, 51, 201, 118, 131, 216, 97, 54].

Comparing QRC("Data") and $QRCS$("Data"), we get that the positions of changed codewords are $[4, 6, 7, 12, 15, 18, 23, 25]$, in which the numbers such as 4 are position numbers of the codewords with underline.

Now we turn to Figure 4(c). In Figure 4(c), the cover message of the QR code is "Escher". The user key and the secret to be hidden are the same as in Figure 4(b) and the mask type is 0. QRC(Escher) = [64, 100, 87, 54, 54, 134, 87, 32, 236, 147, 246, 165, 169, 32, 81, 187, 30, 0, 51, 111, 72, 5, 181, 71, 33, 17]. S and Y are the same as in Figure 4(b). But $S' = [121, 113, 225, 22, 236, 220, 0, 10]$ is different from Figure 4(b). $QRCS$("Escher") = [64, 100, 87, 54, 202, 134, 7,

Table 3: The revealed results of various situations.

Variance	0.1	0.3	0.5	0.7
Gaussian noise				
QR content	Readable	Readable	Readable	Readable
Secret	Decodable	Decodable	Decodable	Decodable
Noise ratio	10%	30%	50%	70%
Salt&Pepper noise				
QR content	Readable	Readable	Readable	Readable
Secret	Decodable	Decodable	Decodable	Decodable
Compression ratio	15%	30%	40%	50%
Jpg compression				
QR content	Readable	Readable	Readable	Readable
Secret	Decodable	Decodable	Decodable	Decodable
broken's blocks	1	2	4	8
Image damage				
QR content	Readable	Readable	Readable	Readable
Rotation angle	45°	90°	180°	270°
Rotation				
QR content	Readable	Readable	Readable	Readable
Secret	Decodable	Decodable	Decodable	Decodable

193, 236, 147, 246, 165, 10, 32, 81, 18, 30, 0, 188, 111, 72, 5, 181, 114, 33, 214]. Comparing *QRC*("Escher") and *QRCS*("Escher"), the position number of the changed codewords also are [4, 6, 7, 12, 15, 18, 23, 25].

In Figure 4(d), under inputting the same key with Figure 4(b), the same secret is hidden in the QR code with the cover message "Linux" and the mask type 0. *QRC*("Linux") = [64, 84, 198, 150, 231, 87, 128, 236, 17, 7, 55, 171, 236, 178, 92, 79, 9, 79, 217, 71, 42, 143, 106, 227, 208, 123]. As the cover message is changed, S' is changed to be [132, 176, 181, 199, 24, 59, 171, 189]. So the final *QRCS*("Linux") = [64, 84, 198, 150, 159, 87, 152, 220, 17, 7, 55, 171, 188, 178, 92, 185, 9, 79, 2, 71, 42, 143, 106, 225, 208, 201] which has different data on the same changed codewords with Figure 4(c).

4.2. Schemes Performance. Considering the noise, compression, damage, etc., caused by the propagation and printing methods in practical applications, we analyzed the performances in Gaussian noise, salt and pepper noise, JPG lossy compression, damage, and rotation of QR codes generated by the proposed scheme hidden secret. As shown in Table 3, the QR code generated by the proposed scheme can be read correctly when being subjected to various distortions.

Table 4 shows a general comparison between the related schemes [6–12, 14, 16, 19] and the proposed scheme. Unlike the conventional hiding and watermarking schemes [6–12, 19], the proposed scheme utilizes the code character of the QR code and embeds the secret into the modules of the QR code directly. Hence, the QR code with the hidden secret information can be easily scanned by barcode readers, which make it suitable to the low-power mobile device applications.

Table 5 shows the tolerant secret capacity of the proposed scheme with different QR versions and error correction levels. The proposed scheme can embed at most $2r$ secret bits into the QR tag. Here, the maximum secret capacity r is decided by the version and error correction level of the QR code. For example, in QR Version 1-H, the proposed scheme can embed at most 64 bits secret into the QR tag.

4.3. Security Analysis. In what follows we discuss the resistance of the proposed scheme to two QR code attacks.

Table 4: Comparison of related QR code schemes.

Methods	[6, 19]	[7, 8]	[9, 11, 12]	[10]	[14, 16]	Proposed
Applications	Image hiding	Image hiding	Watermarking	Watermarking	Secret hiding	Secret hiding
Embedding domain	Frequency	Spatial	Frequency	Spatial	Spatial	Spatial
Computational complexity	High	Low	High	Low	Low	Low
Module-based	No	No	No	No	Yes	Yes
Robustness of secret	High	Low	High	High	High	High
Against contrast analysis attack	-	-	-	-	No	Tes

Table 5: The QR data payload for different QR versions and error correction levels.

Version and Error Correction Level	L	M	Q	H
1	16	32	48	64
10	288	520	768	896
20	896	1664	2400	2800
30	1800	3248	4800	5760
40	3000	5488	8160	9720

Secret hiding payload in schemes [14–16] is not fixed; that is to say, the length of the secret to be embedded (which may be the original secret or the secret with confusion) is decided by the original secret, which result in their schemes being vulnerable to one kind of QR code attacks. Under this attack, when attackers have the information of the version of a QR code, they can generate a QR code in the same version with the same cover message and compare this QR code with the QR code embedding the secret and then get their different part including the secret information. If the payload of the hidden secret is not high, for example, in [15], secret could be leaked when being subjected to brute force attacks. The proposed scheme extends the length of the original secret to the maximum length by adding all 0 sequences to the code of the original secret. After confusion the extended secrets are embedded into the QR code. No matter how long the length of the secret is, the attacks get their different part with the longest length when they compare the original cover QR code with the new QR code embedding the secret. Hence, the scheme has the best resistance to brute force attack even if the original secret is short.

The position and confusion of secret to be embedded are decided only by the user key in schemes [14–16]. Hence, QR codes hiding the same secret with the different cover message and the same key have some same data. These same data can be exploited by contrast analysis attack. The proposed scheme achieves confusing secret using the codewords of the cover message. So when the cover message changes, the data of the changed codewords will be different. For example, as shown in Section 4.1, compared with original QR code in Figure 4(b), QR codes in Figures 4(c) and 4(d) have changed codewords of cover messages on the same position [4, 6, 7, 12, 15, 18, 23, 25] but have different data. Contrasting *QRCS*("Escher") and *QRCS*("Linux"), there are not the same codewords which will leak the secret information. Hence the attacker can not find position where the secret information was embedded by contrasting the code of the QR codes hiding the same secret information with the different cover message and the same key.

5. Conclusion

The QR code secret hiding scheme designed in this paper can hide up to 9720 bits of secret information as needed and does not affect the readability of the cover message. The secret information can be extracted by the authorized user with the right key in the proposed scheme. Hence when the QR code is copied by the attacker, the attacker can not extract the secret without the key. The proposed scheme has low computational complexity and high secret payload and is suitable for low-power devices. In addition, the basic point is that, unlike the other known schemes, the proposed scheme can resist contrast analysis attack, which can prevent forgery if the scheme is applied for e-ticket, copyright protection, and brand anticounterfeit in IoT systems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants nos. 61472472, 61672414, and 61772418, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant no. 2016JM6033, and the Key Laboratory of Applied Mathematics of Fujian Province University (Putian University) under Grant no. SX201807. Qinglan Zhao is supported by the Innovation

Ability Support Program in Shaanxi Province of China under Grant no. 2017KJXX-47.

References

[1] D. Wave, "QR code standardization," 2003, http://www.qrcode.com/en/index.html.

[2] J. Katz, A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, New York, NY, USA, 1996.

[3] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[4] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[5] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, 1999.

[6] W. Y. Chen and J. W. Wang, "Nested image steganography scheme using QR-barcode technique," *Optical Engineering*, vol. 48, no. 5, article no 057004, 2009.

[7] H.-C. Huang, F.-C. Chang, and W.-C. Fang, "Reversible data hiding with histogram-based difference expansion for QR code applications," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 779–787, 2011.

[8] S. Dey, K. Mondal, J. Nath, and A. Nath, "Advanced steganography algorithm using randomized intermediate qr host embedded with any encrypted secret message: ASA_QR algorithm," *International Journal of Modern Education and Computer Science*, vol. 4, no. 6, pp. 59–67, 2012.

[9] M. Sun, J. Si, and S. Zhang, "Research on embedding and extracting methods for digital watermarks applied to QR code images," *New Zealand Journal of Agricultural Research*, vol. 50, no. 5, pp. 861–867, 2007.

[10] M. Gao and B. Sun, "Blind watermark algorithm based on QR barcode," in *Foundations of Intelligent Systems*, vol. 122 of *Advances in Intelligent and Soft Computing*, pp. 457–462, Springer, Berlin, Germany, 2012.

[11] S. Rungraungsilp, M. Ketcham, V. Kosolvijak, and S. Vongpradhip, "Data hiding method for QR code based on watermark by compare DCT with DFT domain," in *Proceedings of the 3rd international conference on computer and communication technologies*, pp. 144–148, India, 2012.

[12] L. Li, R. Wang, and C. Chang, "A digital watermark algorithm for QR code," *International Journal of Intelligent Information Processing*, vol. 2, no. 2, pp. 29–36, 2011.

[13] P.-Y. Lin and Y.-H. Chen, "QR code steganography with secret payload enhancement," in *Proceedings of the 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–5, 2016.

[14] Y.-J. Chiang, P.-Y. Lin, R.-Z. Wang, and Y.-H. Chen, "Blind QR code steganographic approach based upon error correction capability," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 10, pp. 2527–2543, 2013.

[15] P.-Y. Lin, Y.-H. Chen, E. J.-L. Lu, and P.-J. Chen, "Secret hiding mechanism using QR barcode," in *Proceedings of the International Conference on Signal-Image Technology Internet-Based Systems*, pp. 22–25, 2013.

[16] P.-Y. Lin and Y.-H. Chen, "High payload secret hiding technology for QR codes," *Eurasip Journal on Image and Video Processing*, vol. 2017, no. 1, article no 14, 2017.

[17] T. V. Bui, N. K. Vu, T. T. Nguyen, I. Echizen, and T. D. Nguyen, "Robust message hiding for QR code," in *Proceedings of the Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 520–523, IEEE, Kitakyushu, Japan, August 2014.

[18] D. Wave, "Information technology automatic identification and data capture techniques QR code bar code symbology specification," in *Proceedings of the International Organization for Standardization, ISO/IEC*, vol. 18004, 2015.

[19] C. Chung, W. Chen, and C. Tu, "Image hidden technique using QR-barcode," in *Proceedings of the Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 522–525, Kyoto, Japan, September 2009.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 1694702, 12 pages
https://doi.org/10.1155/2019/1694702

Research Article

Fingerprint Protected Password Authentication Protocol

Chao Yang[1,2] **Junwei Zhang**[1] **Jingjing Guo**[1] **Yu Zheng**[1] **Li Yang**[3] **and Jianfeng Ma**[1]

[1]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*
[2]*Science and Technology on Communication Networks Laboratory, Shijiazhuang 050081, China*
[3]*School of Computer, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Chao Yang; chaoyang@xidian.edu.cn

Received 12 March 2019; Accepted 22 May 2019; Published 26 June 2019

Guest Editor: Fagen Li

With the rapid development of industrial Internet of things (IIOT), a variety of cloud services have been deployed to store and process the big data of IIOT. The traditional password only authentication is unable to meet the needs of security situation in IIOT. Therefore, a lot of mobile phone assisted password authentication schemes have been proposed. However, in existing schemes, the secret information is required to be stored in the user's mobile phone. Once the phone is lost, the secret information may be obtained by the opponent, which will bring irreparable loss to the user. To address the above problems, we propose a fingerprint protected password authentication scheme which has no need to store the secret parameter in the mobile phone. When a user logs in, he uses his mobile phone to generate the private key which is used to decrypt the encrypted text generated during the registration phase. The process of generating the private key needs to enter the password and the fingerprint. When the computer interacts with the mobile phone, the user's password will be blinded so that it can protect the user's password from adversary's attacks. Theoretical analysis and experimental results show that our scheme improves the security of the user's secret. Meanwhile, our scheme can resist the opponent's dictionary attacks, replay attacks, and phishing attack. Our scheme can reduce the storage pressure of the mobile phone and is easy to deploy.

1. Introduction

The rapid development of industrial Internet of things (IIOT) has reformed many aspects of user's daily life. A variety of cloud services have been deployed to store and process the big data of IIOT [1–4]. Due to the openness of public network, these cloud services suffer from a wide range of attacks [5–8]. Text-based passwords remain the dominant authentication technique for various online services and systems, because it is accessible and convenient. Most of them require their own password and the user has to manage multiple passwords to their accounts.

However, password authentication schemes have inherent limitations. Firstly, the user is prone to use a simple password for convenience. If the length of the password is short, its information entropy will be very low. Therefore, the user's password will be easily compromised by attackers. After obtaining the user's password, the opponent can use the password to access the user's account and steal user's private information. In addition, the user may use a single password to authenticate to multiple services. If one of the user's accounts information leaks, it would pose a threat to the security of other accounts. What is worse, users often forget their passwords and try to login via trial-and-error, which means that a malicious online service would learn not only a user's password to that service, but to many other services, possibly also through a cross-site impersonation attack. The low information entropy and the repeated setting of the password are common problems in the process of password setting and password usage. These above problems are also easily exploited by attackers. Moreover, malicious service providers may also launch attacks against the user. For example, the CEO of Facebook allegedly used Facebook login data to access the private mails of some business rivals and journalists in 2004[9]. Hence, protecting users' passwords has become extremely important for individuals.

To avoid the limitation of password authentication mechanism, researchers have proposed many schemes [10–12]. Adding an additional factor is one of the common ways to protect user's password. With the extensive use of mobile phones in recent years, users often use mobile phone to manage personal information and store many credentials or secrets on the mobile phone. Consequently, the research on mobile phone assisted password authentication is one of the important trends.

In 2004, Wu et al. [13] proposed a scheme using the mobile phone as an authentication token with the help of a trusted agent P. When a user wants to authenticate to a remote server, he/she is required to communicate with the agent P first. The agent P sends a session name to the mobile phone and the PC client. The user compares the session name showing on the mobile phone and the PC client. Then the user sends the result to the agent P. However, this scheme requires the agent to be secure and trustworthy, which is a strong assumption in practice. Moreover, the mobile phone is required to interact with the agent several times and is prone to man-in-the-middle attack. In addition, the involvement of a trust agent makes the scheme hard to be deployed.

The scheme [14] proposed by Thanh et al. uses a mobile phone and an authentication server to assist the user during login authentication. In this scheme, with the help of the mobile phone and the authentication server, the client and the server is not required to communicate directly through the public network while sending or receiving credentials. However, the authentication server is utilized to receive the data transmitted by the mobile terminal through the GSM network, which is considered to be an insecure transmission channel. Therefore, the response phase is prone to be attacked by the opponents. When the user uses a mobile phone to send a message to the authentication server through the GSM network, it is not hard for the opponent to hijack the user's message. The adversary can use the hijacked information to authenticate with the server.

Considering the security of information transmission in the authentication phase, the scheme in [15] uses the mobile phone as the intermediary to transmit the user's secrets securely. It uses the QR code and the camera to encrypt the key during the authentication phase. But this scheme requires adding a bar code on the device in advance. If there are plenty of authentication devices or services, many QR codes are required to be predeployed, which is not practical for deployment.

The scheme in [16] is put forward to resist session hijacking and phishing attacks. The client PC is independent to the mobile device and just performs calculations in the scheme, while the mobile phone records the user's password. They share a session key to against session hijacking. However, the mechanism requires the terminal to calculate and establish a pair of public keys with each server. If there are a number of services, the setting will be cumbersome, which will bring much overhead to the mobile phone. What is more, this scheme needs to establish a secure channel between the client PC and server, which greatly reduces the availability of this scheme.

In [17], a method for users to authenticate with the server on an unreliable computer is provided. The main idea is that the mobile phone scans the QR code on the computer, encrypts the QR code, and then uses it to authenticate with the server. However, this scheme requires the trusted mobile phone and the server to set up a shared key in advance, which is difficult to realize in the practice. Moreover, mobile phone and computer need to communicate with the server several times during the authentication phase, which makes this scheme time consuming.

In the SPA (single password authentication) scheme proposed by Acar et al. [18], it requires additional cloud storage to store the user's data and assist the user to register and log in. However, the scheme needs to set a secure channel between the client and the storage when the client sends the blind signature private key to the storage, which is a strong assumption. Furthermore, this scheme also requires a trusted cloud storage to store the ciphertext, which is not readily accessible in practice. Moreover, the user and the cloud storage require a lot of data transmission and interaction during the authentication phase, making time consuming. Acar et al. also proposed a different SPA scheme based on a mobile device. In this scheme, a trusted mobile device is used to assist users to register and log in. The user uses the password to encrypt the authentication key and then store the ciphertext on the trusted mobile phone device. When the user wants to access his account, he inputs his password to decrypt the ciphertext to get his authentication key and then authenticate with the server. But this scheme also requires the mobile device and server to be trusted, and no collusion between the server and mobile phone, which is a very harsh security requirement for mobile devices. In addition, if the mobile device is lost, which is a very common situation, user's ciphertext stored on the mobile device can be cracked offline by adversaries, which will lead to serious consequences.

In the password authentication method based on mobile phone assistance, there is a method based on fingerprint, which attracts much attention. The scheme [19] proposed using a fingerprint to replace the PIN code to authenticate the user's identity. However, fingerprint information is still stored in the mobile phone in this scheme. If the fingerprint information is cracked by adversaries, it will pose a security threat to the user. So this scheme cannot protect the user's privacy very well. The scheme [20] proposes a solution that enables a user to authenticate with a remote server through a password and fingerprint. However, this scheme requires a secure channel between the client and the server, which is difficult to deploy in the actual scenario. Moreover, the mobile device is also required to be trusted. It means that if the mobile device is lost, the ciphertext and the fingerprint parameters stored on the phone can be cracked by attackers.

The password authentication method based on mobile phone assistance has received extensive attention and has been widely studied. However, these schemes have security risks in terms of user's privacy: the mobile phone has to store the user's privacy or credentials during the authentication process and the mobile phone must be trusted. Otherwise the user's privacy stored in the mobile phone can be cracked forcedly by the adversary. In March 2016, FBI announced that

it can crack the IOS to obtain the suspect's mobile phone terminal information [21]. This means that there is no mature technology that can fully guarantee the user's privacy stored in the mobile devices at present. So it is risky to store the user's secret information on the mobile phone. This risk will affect the security of the authentication scheme based on mobile phone assistance extensively.

Although authentication schemes assisted by mobile phone are popular, there are a lot of defects and hidden danger in this kind of authentication schemes. At present, most authentication schemes with mobile phone assistance have to store the user's private information on the mobile phone to help authentication. Moreover, in most schemes, the mobile phone is required to be fully trusted. However, in March 2016, FBI announced that it can crack the IOS system to obtain the suspect's mobile phone terminal information. This means that there is no mature technology that can fully guarantee the user's privacy stored in the mobile devices at present. So it is risky to store the user's secret information on the mobile phone. HCR (Hidden Credential Retrieval from a Reusable Password) [22] is a protocol that can be used by the user to store his information on an unreliable remote server. When the user wants to obtain his information, he can get his own encrypted information by a preset password and do not have to disclose the password to the server. However, this protocol needs to set up a secure channel to transmit the private key during the registration phase. Meanwhile, the private key has been saved on the server for a long time, which is an insecure operation. Existing three party authentication schemes participated with mobile phone, computer, and server often require the user's encrypted information saved on the mobile phone; meanwhile, the mobile phone must be reliable. What is more, the scheme assumes a secure channel to transmit secret information. This assumption and deployment are too difficult to be implemented in an actual situation, which greatly reduces the availability and security of the authentication scheme.

To solve these difficult problems, we propose a fingerprint protected password authentication (FPPA). Our new scheme has eliminated the complete credibility of the phone and the security channel between the phone and the server. When the user logs in he uses his mobile phone to generate the private key which is used to decrypt the ciphertext generated during the registration phase. The user needs to enter his password and fingerprint at the private key generation process. When the computer interacts with the mobile phone, the user's password will be blind. So the password can be protected from adversaries' attacks. Our scheme does not need to store any user's secret information on the mobile phone. Even if the mobile phone is stolen by adversaries, the private information in it will not be leaked. Security analysis proves that this scheme can protect the user's password against dictionary attacks and session hijacking attacks even without the assumption of secure storage and a secure channel between phone and server. Experimental results show that our scheme is essentially the same as the current schemes in terms of performance and has a significant advantage in the storage of mobile terminal over other schemes. The main contributions of our work can be summarized as follows:

(I) In order to ensure the security of the user's secret information in the IIOT, a fingerprint protected password authentication protocol which has no need to store the secret parameter in the mobile phone is proposed.

(II) Our scheme can resist the opponent's dictionary attacks, replay attacks, and phishing attack.

(III) Our scheme has good performance and strong practicability.

The rest of the paper is organized as follows. Section 2 introduces the preliminary knowledge of the paper. Section 3 introduces the model of the system model and the adversary model respectively. Section 4 introduces the overview of the FPPA scheme. Section 5 describes the security analysis and proof. Performance analysis and evaluation are presented in Section 6. Finally, Section 7 concludes.

2. Preliminary

2.1. Blind Signature. Blind signature scheme is a basic cryptographic primitive to guarantee the anonymity of participants. A blind signature scheme consists of two entities: a message sender and a signer. It allows the sender to obtain the signature of a given message without revealing any information about the message and the corresponding signature. The basic principle of blind signature is the application of two commutative algorithms. One algorithm is to conceal information, called blind transformation, and the other is the signature algorithm.

The characteristics of blind signature are as follows:

(I) The content of the message is blind to the signer.

(II) The signer cannot associate the signed message with the actual message signed. Even if he saves all the documents he signed, he cannot distinguish the real content of the documents he signed.

2.2. HKDF (Halting Key Derivation Functions). HKDF [23] is a protocol that can generate strong security key through a general weak password. It can choose the cost of the calculation, which is applied to the password-based encryption system.

HKDF includes two algorithms: *HKDF.prepare*() and *HKDF.extract*().

(I) $HKDF.prepare(w, t, r) \longrightarrow y, v$. *HKDF.prepare*() takes as input a password w, a random string r, and a parameter t of cycle calculations and returns a random token y and its ciphertext v.

(II) $HKDF.extract(w, v) \longrightarrow y$. *HKDF.extract*() takes as input a password w and a ciphertext v and either returns a token y or fails to halt in polynomial time. If the password w is not correct, this algorithm will always be a loop operation without output feedback 0.

In this paper, we use a *HKDF* algorithm to generate a strong key using a private key, and user or system can establish a suitable parameter value to select the corresponding calculation cost.

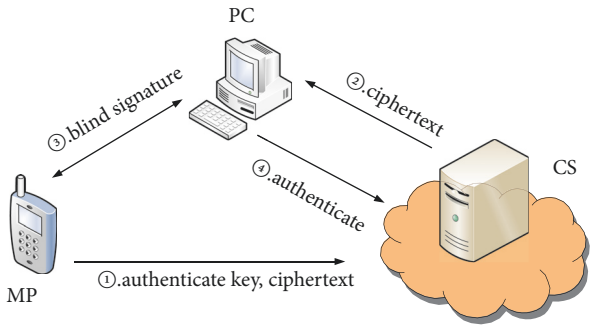


Figure 1: System architecture of our scheme.

3. System and Adversary Model

3.1. System Model. The system model of the scheme is shown in Figure 1. It mainly includes the following three entities: mobile phone (MP), personal computer (PC), and cloud server (CS).

MP is in charge of user's authentication key and generation of the ciphertext during the registration phase. It interacts with the PC in the authentication phase, so that the user's blind password and fingerprint parameters are processed during this phase. In addition to the basic functions of the mobile phone, it should be able to register the user's fingerprint data and connect to the PC via Bluetooth, USB, or Wi-Fi, in order to communicate with PC.

PC is responsible for the password input and interacts with mobile phone to generate session key when the user logs on. PC should ensure that it is connected to the Internet and can communicate with the CS and mobile phones.

CS stores users' authentication key and their ciphertext at the end of the registration phase and provides users with their corresponding ciphertext information. Meanwhile, it verifies user's identity in the login phase. Accordingly, the CS should be able to connect to the Internet and has a powerful database to store user's information.

The registration phase includes step ①, and the login phase includes steps ②, ③, and ④.

3.2. The Adversary Model. The adversary model is defined as follows:

(I) PC is distrusted. In the login phase, if the user's PC is infected by malwares, the keyboard may record the user's password. If the user accesses a phishing site, the adversary may also record the user's password.

(II) User uses mobile phone during the registration and login phase, while the mobile phone is prone to be lost or be implanted with malwares. If the phone was infected by malwares, the adversary may impersonate as the user to communicate with the PC.

(III) The CS authenticates the user and provides application services. In the registration phase, the CS is trusted. After that, the CS may suffer from replay attack or dictionary attacks. In addition, the CS may also be curious about user information.

4. The FPPA Scheme

4.1. Design Rationale of the FPPA Scheme. Our FPPA protocol allows a user register and authenticate with online services using his password and mobile phone securely. During the authentication protocol, the mobile phone may be lost or compromised by malwares, so the information stored on the mobile phone may be exposed by adversaries. At the same time, when the user logs on, the mobile phone interacts with the PC. If the mobile phone gets the user's password, it may also be stolen by the adversary. Therefore, it is required that authentication information and encrypted data should not be stored in the mobile phone. To avoid the case that the password is stolen by adversaries during the authentication phase, it is required that the PC encrypts the password after the user enters his password.

To this end, we improve the existing HCR protocol, which is deployed on the PC and mobile phone, such that user can interact with the distrusted mobile phone. In the HCR protocol, the private key needs to be transmitted in a secure channel and stored safely for a long time. Our scheme replaces the private key parameter with the user's fingerprint so that our scheme does not require the secure channel, or the long-term storage.

4.2. The Detailed Process of the FPPA Scheme. Our scheme includes a mobile phone, a PC, and a CS during the registration phase and the login phase. The user enters his password and fingerprint on the phone and PC to register and log on. The CS is responsible for the registration and authentication when the user wants to access the CS.

The system initialization definition is as follows:

p: a prime number;

name: the user name;

pwd: the user's password;

G: a cyclic abelian group of order p;

F: a multiplicative domain of order p;

e: the user's fingerprint;

s: the user's private key;

y: the user's authentication key;

v: the ciphertext of the authentication information;

$Hash() : \{0,1\}^* \longrightarrow G$: a cryptographic hash function, which is to be viewed as a random oracle;

$d \in F_p^x$: a random number generated by the system;

f: a hash function;

HKDF: a key derivation function, which consists of two functions: *HKDF.prepare*() and *HKDF.extract*().

Generally, in the previous mobile phone based authentication scheme, the mobile phone stores user's authentication information. The user's password is encrypted and stored on the mobile phone, or the user's secret key is encrypted with his password and stored on the mobile phone. If the phone is compromised by the adversary, the adversary is likely to get the user's password or authentication information.

In our FPPA scheme, the mobile phone does not store the user's authentication information. The authentication key is regenerated by himself when he wants to log on. Even if the device is stolen, it will not pose threat to the user's password,

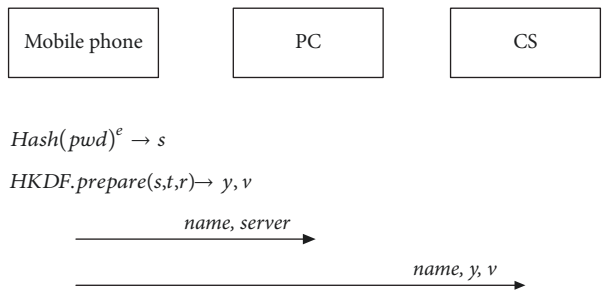


Figure 2: Registration phase.

fingerprint, or encrypted information. Moreover, when the user sends the name to the server in the registration phase, he/she can also choose to send a random identity ID, so that he/she can preserve his/her anonymity. Even if several CSs are in collusion with each other, the user cannot be attacked.

Registration and login phase are as follows:

4.2.1. Registration Phase. In this phase, before the user accesses to the service provided by the CS, he/she is required to register to be a legal user. The user can register through the mobile phone, after the completion of registration phase, the user name *name* and service information are sent to the PC side.

The process is detailed as follows, as is shown in Figure 2.

(I) Mobile phone generates the user's private key s according to the user's password *pwd* and fingerprint parameter *e*. The user's private key *s* is generated as follows:

$$Hash(pwd)^e \longrightarrow s. \quad (1)$$

(II) The mobile phone side generates the user's authentication key *y* and the ciphertext of authentication key *v* based on the HKDF function and the private key *s*. The operation is as follows:

$$HKDF.prepare(s,t,r) \longrightarrow y, v, \quad (2)$$

where *r* is a random string generated by the system; *t* is the number of loop operations. Note that *t* can be selected according to the security requirements of the user. If the value of *t* is larger, the computation would be more time consuming. So the generated key is more secure.

(III) The mobile phone sends the authentication key *y*, the ciphertext *v*, and the user name *name* to the CS. Meanwhile, the mobile phone sends the user name *name* and CS information *server* to the PC. The user just remembers the user name *name* and password *pwd*, and the mobile phone does not store any secret information of the user.

4.2.2. Login Phase. When the user wants to access the CS, he/she needs to send a login request to the CS. The user enters his own password and fingerprint data to generate a private key and then decrypts the ciphertext to obtain the authentication key. The server determines whether to accept the logon request or not after the server computes and compares the authentication information. The specific operation is as follows, as is shown in Figure 3

(I) The PC sends *name* to the CS, and then the CS retrieves the corresponding information according to *name*. After that, the CS sends the corresponding ciphertext *v* and certified random number *chal* to the PC.

(II) The user enters his password on the PC. The system generates a blind parameter *d* and then calculates the blinded password μ to the mobile phone.

$$Hash(pwd)^d \longrightarrow \mu \quad (3)$$

(III) The user inputs his fingerprint on the mobile phone. The mobile phone gets the fingerprint parameter ε and then obtains β by signing the blinded password μ. The mobile phone sends β to the PC.

$$\mu^{\varepsilon} \longrightarrow \beta \quad (4)$$

(IV)The PC recovers the private key *s* by the following calculation.

$$\beta^{1/d} \longrightarrow s \quad (5)$$

(V)The PC uses the private key *s* and HKDF algorithm to decrypt the ciphertext *v* and then obtains the authentication key *v*.

$$HKDF.extract(s,v) \longrightarrow y \quad (6)$$

(VI)The PC obtains the response by computing $f(y, chal) \longrightarrow response$ and sends the response *response* to the CS.

(VII)Finally, the CS computes $response' \longleftarrow f(y, chal)$ and compares it with the response received. This logon is successful if these two parameters are identical.

5. Security Analysis and Proof

5.1. FPPA Protocol. Our FPPA protocol has three types of entities: the PC client who wants to use a password and fingerprint to access services, the CS who registers and authenticates clients, and mobile phone who assists PC client to complete registering and logging.

Our FPPA protocol consists of following algorithms:

UserGen. This algorithm is run by the user to generate a user name *name* and an l-bit password *pwd*.

Register. The user registers with the server by inputting his password and fingerprint on the mobile phone. In the end, the mobile phone outputs an authenticated key *y* and ciphertext of the authenticated key *v*. The mobile phone sends (*name*, *y*, *v*) to the CS and sends (*name*, *server*) to the PC.

Store. The PC stores the user name *name*. The mobile phone does not store user's any information. The user just needs to remember his name and password.

Retrieve. The PC client uses its user name *name* to retrieve its ciphertext *v* from the server. The server sends the ciphertext *v* and challenge *chal* to the PC client.

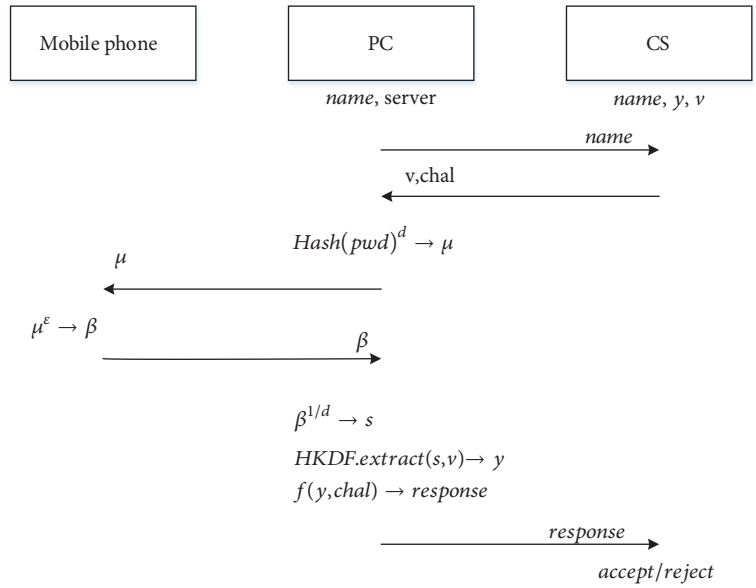


Figure 3: Login phase.

PreAuth. The user inputs his password *pwd* in the PC side and inputs his fingerprint *e* on the mobile phone. The PC client interacts with the mobile phone using the blind signature. At the end of the blind signature, the PC client generates the private key *s* and decrypts the ciphertext *v* using *s*. The PC client obtains the authenticated key *y* after decrypting the ciphertext.

Authenticate. The user uses his authenticated key *y* and change *chal* to prove to the server that he owns the corresponding account credential. Finally, the CS outputs accept or reject.

5.2. Security Game. Our FPPA scheme assumes that the user has a mobile phone and PC client, so that the user can enter his own password and fingerprint on these devices. However, mobile phones may be lost. Therefore, malicious adversaries may use the user's data stored on the mobile phone to access the server, and even decrypt the ciphertext brutally. What is more, the PC client and the server are also easily attacked by adversaries [24, 25]. We divide the potential attacks into two types: Insider attack (Game One) and Outsider attack (Game Two). Let x be a security parameter, such that all hash functions have at least l≥2x bits of output. Let $|D| << 2^x$ be the length of the password dictionary. k is the length of the user's password. $k \in N$. Let neg(k) be a negligible function of k if ∃K finite:∀k>K: $neg(k) < k^{-c}$[26–28].

Game One (Outsider Attack). In this game, the adversary can control the PC side. He can perform the Store step with the mobile phone and PC. He can also perform the *PreAuth* step with the server by acting as a malicious PC. The adversary may also control the PC side and then launch the Retrieve request. We assume the times of requests are q. The adversary can control the mobile phone side and perform Retrieve step with the PC side by simulating the mobile phone side. In the algorithm of our scheme, mobile phone can only receive data from the PC side but cannot initiate requests to the PC side. If the adversary got the user's password through the game or guessed the private key s without the password, we consider the adversary win [29, 30].

Definition 1. We assume that the probability of adversary winning the Game One is P1. If P1≤max[q/|D|+neg(k), 1/2|s|], we consider that FPPA is in accord with the security based on Game One.

Game Two (Insider Attack). In this game, the adversary can control the server side. He can launch an offline dictionary attack on the server to get the ciphertext of the user's password. We assume that the adversary can initiate *T* times offline dictionary attacks against *HKDF.extract*. If the adversary got the user's password through the game, we consider the adversary wins.

Definition 2. We assume that the probability of the adversary winning the Game Two is *P2*. If $p2 \le q/|D|+2T/|D|t+neg(k)$, we consider that FPPA is in accord with the security based on Game Two.

Definition 3. If FPPA is in accord with the security based on Game One and Game Two, we consider the FPPA is security.

5.3. Security of FPPA

Theorem. *FPPA is in accord with the security based on Definition 3.*

Prove. (I) FPPA can provide security based on Game One.

Game One. (1)The adversary simulates the PC side and performs *Store* with mobile phone to get the user's name *name*. The adversary performs *PreAuth* with server side and retrieves user secret key ciphertext v and challenge *chal* according to the user's name. According to v is generated by the algorithm of *HKDF* [23], if someone inputs wrong password to decrypt the ciphertext, the *HKDF* would compute circularly with no output. So the adversary cannot get user's password through v. Consequently, *P1* is a small value that can be ignored.

(2)The adversary simulates the PC side, generates a password *pwd0*, and performs *Retrieve* with mobile phone. This interaction process is similar to the interaction process in the algorithm of *HCR* and just replaces the signature private key in *HCR* into fingerprint parameters. The probability of guessing password in this game is the same as that in HCR [22]. So the probability of the adversary guessing the password: $P[\text{guess password}] \leq q/|D| + neg(k)$. The probability of the adversary winning this game: $P1 \leq q/|D| + neg(k)$.

(3)The adversary controls the PC side and guesses the private key s without password. The probability of the adversary guessing private key: $P[\text{guess s}] \leq 1/2^{|s|}$. The probability of the adversary winning this game: $P1 \leq 1/2^{|s|}$.

(4)The adversary performs *Retrieve* with PC side by simulating mobile phone. The adversary can get the blinded password of the user. So the probability of the adversary guessing the password is the same as the analysis of (b). The probability of the adversary guessing the password: $P[\text{get the password}] \leq q/|D| + neg(k)$.The probability of the adversary winning this game: $P1 \leq q/|D| + neg(k)$.

In summary, $P1 \leq \max[q/|D| + neg(k), 1/2^{|s|}]$. So FPPA can provide the security based on Game One.

(II) FPPA can provide security based on Game Two.

Game Two. Malicious server launches offline dictionary attacks on the ciphertext stored on itself. It generates a private key s' and tries to decrypt ciphertext: *HKDF.extract (s',v)*. According to the security of *HKDF*[23]:

$P[getthepassword] \leq T/|D|t + neg(k)$ .t is the parameter of cyclic operation times set up in *HKDF.prepare*.

So the probability of the adversary getting the user's password: $P2 \leq q/|D| + 2T/|D|t + neg(k)$.

Consequently, FPPA can provide the security based on Game Two.

In summary, FPPA can provide security based on Game One and Game Two, so FPPA is a secure protocol.

6. Performance Analysis and Evaluation

6.1. Test Plan and Scenario. We rented Ali CS to authenticate with the user in order to make the test scenario more close to the actual one. Ali CS deployed in Qingdao and the distance between Qingdao and Xi'an is 1058.6km.

As shown in the Figure 4, 1 describes that the mobile phone computes and generates authenticate key y and authenticate key ciphertext v. And then the mobile phone sends v to the CS in the end of registration phase. 2, 3, and 4 describe that the mobile phone and the PC compute and generate authenticate key y to authenticate with CS in login phase.

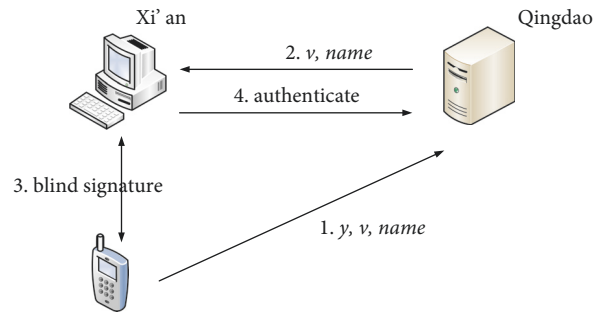


Figure 4: The overall test scenario.

Android device, a PC and CS are used to simulate our scheme. We evaluate the usability of our scheme from the aspects of time and storage. We test the user's registration and login time in four different scenarios. In terms of storage capacity, we mainly analyze and evaluate the storage capacity of the Android device. The experimental devices of our scheme are a PC, a CS, and an Android device. The specific parameters of the experimental devices are as follows:

We use Ali CS located in Qingdao as the CS in our scheme. RAM of the CS is 1GB. It has a single-core processor. The operating system of it is windows Server 2008.

We use HP Compad dx7408 MT DT PC as the computer in our scheme. It has a dual-core processor. RAM of the computer is 2GB. The operating system of it is windows 7(32 bits).

We use Bluestacks (Android 4.1.2-API Level 16, CPU ARM (armeabi), RAM 512, VM Heap 16, Internal Storage 200MiB) and Android phone as the mobile phone in our scheme.

We tested our scheme in 4 cases.

Case 1. The user registers and logs in, and then enters his password and fingerprint. We test the time spent on registration phase and login phase and then test several sets of data to find the average time for one user spent on registration and login phase. The Android device used in this scenario is the Android simulator.

Case 2. The length and complexity of passwords are different for different users. In order to analyze the influence of different passwords on our scheme, we select 20 groups of users. We test the time of these users spent on registration phase and login phase. The Android device used in this scenario is the Android simulator.

Case 3. Due to the different system settings, different length of fingerprint parameters is generated in this case. We test the time spent on registration phase and login phase while the length of the fingerprint is 128b, 256b, and 512b. The Android device used in this scenario is the Android simulator.

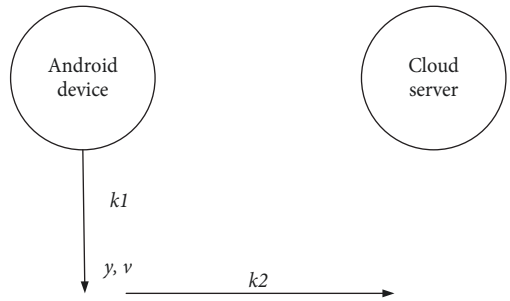


Figure 5: Registration phase.

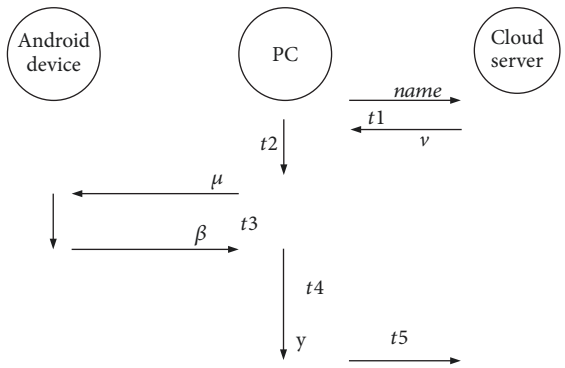


Figure 6: Login phase.

Case 4. The user uses different Android devices (Android emulator, Meizu mobile phone, and HTC mobile phone) for registering and logging in.

We test the time spent on registration phase and login phase. We divide the total time of registration phase K into two parts: $k1$ and $k2$ in this experiment. As shown in Figure 5, $k1$ms is the time spent on the registration phase starting from the user's registration on the Android device to the end of the generation of authenticated key ciphertext. $k2$ms is the time spent on sending the authenticated key to the CS.

The total time T ms of the login phase is divided into five parts. As shown in Figure 6, t1 is the time spent on requesting the corresponding authentication key ciphertext from the user's PC. t2+t4 is the time spent on blinding the user's password and generating the authentication key. t3 is the time spent on signing the blinded password with fingerprint data on the Android device. t5 is the time spent on sending the user authentication information to the CS.

6.2. Experiment Data

6.2.1. Experiment Data of Case 1

(I) Registration Phase. In this case, the user enters the user name, password, and fingerprint on the Android device for registration. The length of user's fingerprint is 256b. As shown in Table 1, we measured 10 groups of data. We can calculate that the average registration time is 216.9ms through the following data.

Table 1: Registration phase time of Case 1.

Test Group	Registration Phase Time/ms		
	k1	k2	K
1	197	19	216
2	200	18	218
3	199	18	217
4	210	17	227
5	199	15	214
6	198	18	216
7	196	17	213
8	190	19	209
9	213	17	220
10	199	20	219

Table 2: Login phase time of Case 1.

Test Group	Login Phase Time/ms				
	t1	*t3*	*t2+t4*	*t5*	*T*
1	39	90	36	19	184
2	45	98	39	18	200
3	38	94	41	18	191
4	38	92	38	16	184
5	37	110	40	17	204
6	39	98	39	19	195
7	40	96	35	17	188
8	41	97	41	19	198
9	39	92	40	18	189
10	38	99	38	17	192

(II) Login Registration. When the user logs in, he enters his password and fingerprint on the PC and Android emulator. We measured the time spent on each stage and then measured 10 groups of data showing in Table 2. We can calculate the average login time is 192.5 ms through these following data.

6.2.2. Experiment Data of Case 2

(I) Registration Phase. In this case, we test 20 groups of different users. The user inputs his name, password, and fingerprint on Android emulator for registration. We measure the time (k1, k2) spent during the registration phase. The complexity and length of the user name and password are different. The length of these users' fingerprint is 256 bits. The measured result is shown in Table 3.

(II) Login Phase. Users enter their password and fingerprint on the PC and Android devices. We measure the time spent on each stage. 20 groups of data are shown in Table 4.

6.2.3. Experiment Data of Case 3

(I) Registration Phase. The user inputs his password on Android device, and then the Android device generates three kinds of fingerprint data with different length (512b, 256b,

Table 3: Registration phase time of Case 2.

User	Registration Phase Time/ms		
	k1	*k2*	*K*
User1	222	15	237
User2	195	17	212
User3	271	17	288
User4	236	19	255
User5	243	20	263
User6	190	17	207
User7	220	18	238
User8	214	16	230
User9	195	21	216
User10	239	18	257
User11	197	16	213
User12	220	19	239
User13	230	18	248
User14	220	17	237
User15	200	18	218
User16	261	19	280
User17	199	16	215
User18	215	18	233
User19	250	19	269
User20	234	17	251

Table 4: Login phase time of Case 2.

User	Login Phase Time/ms				
	t1	*t3*	*t2+t4*	*t5*	*T*
User1	45	88	36	17	186
User2	37	94	38	17	186
User3	37	93	41	15	186
User4	38	63	35	18	154
User5	37	90	41	20	188
User6	39	111	39	19	208
User7	37	93	42	17	189
User8	40	98	36	16	190
User9	39	90	38	21	188
User10	37	92	38	19	186
User11	38	94	42	17	191
User12	40	80	35	18	173
User13	37	98	36	19	190
User14	39	91	44	18	192
User15	36	89	36	19	180
User16	38	92	40	17	184
User17	37	100	39	20	196
User18	38	93	37	19	187
User19	39	92	40	17	188
User20	37	94	39	18	188

and 128b). We measure the time (k1, k2) spent during the registration phase. The measured result is shown in Table 5.

Table 5: Registration phase time of Case 3.

Length Of Fingerprint/b	Registration Phase Time/ms		
	k1	*k2*	*K*
512	210	19	229
256	205	18	223
128	192	18	220

Table 6: Login phase time of Case 3.

Length Of Fingerprint/b	Login Phase Time/ms				
	t1	*t3*	*t2+t4*	*t5*	*T*
512	39	100	39	18	196
256	39	98	38	18	193
128	38	95	38	17	186

(II) Login Phase. The user inputs his password and fingerprint on PC and Android devices. We test the time spent during the login phase. The measured result is shown in Table 6.

6.2.4. Experiment Data of Case 4

(I) Registration Phase. In this case, there are three different Android devices (Android simulator, Meizu phone, and HTC phone). The user enters his password and fingerprint on different Android devices. We measure the time spent during this phase (*k1, k2*).

(II) Login Phase. The user enters his password and fingerprint on PC and different Android devices. We test the time spent during the login phase. The measured result is shown in Table 8.

6.3. Performance Analysis. We can calculate the average time the user registered on the Android simulator is 216.9ms according to the Table 1. The average time required for the user to log in using the Android emulator and PC is 192.5ms.

As shown in Table 3, when different users enter a different password to register, the difference between the longest and the shortest time is 81ms which is less than the human reaction time. Users cannot feel the difference in the length or complexity of passwords. Therefore, the length and complexity of the password have little impact on the user's registration time. The comparison of the time spent by different users in the registration phase is shown in Figure 7.

As shown in Table 4, when different users enter their password to login, the difference between the longest and the shortest time is 54ms, which is less than the human reaction time. Therefore, the length and complexity of the password have little impact on the user's login time. The comparison of the time spent during the login phase is shown in Figure 8.

When the length of the fingerprint is different, the maximum difference of the time among users during the registration phase is 9ms. The maximum difference of the time spent in the login phase is 10ms which has little effect on users. When the length of the fingerprint is 128b, 256b,

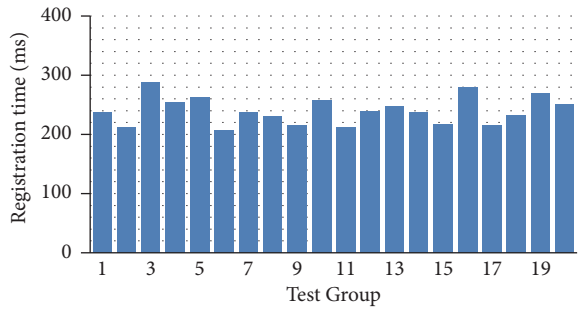


Figure 7: Registration phase experimental data of Case 2.

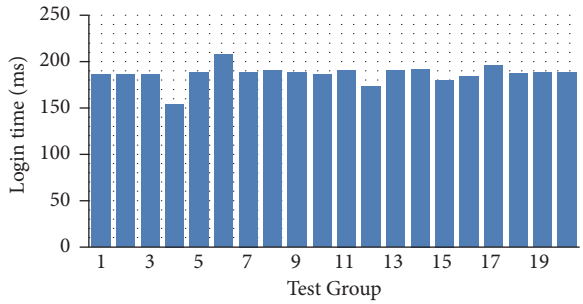


Figure 8: Login phase experimental data of Case 2.

Table 7: Registration phase time of Case 4.

Device	Registration Phase Time/ms		
	k1	*k2*	*K*
Bluestacks	200	17	217
MEIZU	18	19	37
HTC	19	21	40

Table 8: Login phase time of Case 4.

Device	Login Phase Time/ms				
	t1	*t3*	*t2+t4*	*t5*	*T*
Bluestacks	39	96	39	18	192
MEIZU	40	56	35	21	152
HTC	38	57	36	20	151

and 512b, the comparison of the time spent in the registration phase and the login phase is shown in Figure 9.

In Case 4, the user enters the same password and fingerprint on different Android devices. As shown in Tables 7 and 8, the total time spent on registering and logging in on mobile devices is less than the total time spent on the Android simulator. There is no difference in their transmission time. The main difference is the calculation time on Android devices. In the existing Mobile SPA scheme, the total time (26ms) spent in the registration phase is the transmission time and the computation time on the mobile terminal. The time spent on the two mobile phones in this paper is 37ms and 40ms. The difference of the time between them is small, so it has little effect on the performance of the scheme. In the SPA Mobile scheme, the time (38ms) spent in the login phase

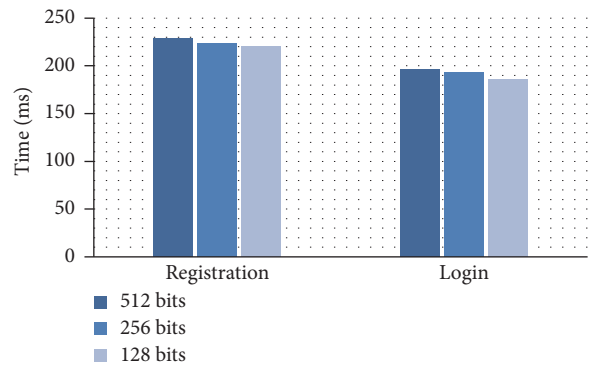


Figure 9: Comparison of experimental data in two phases of Case 3.

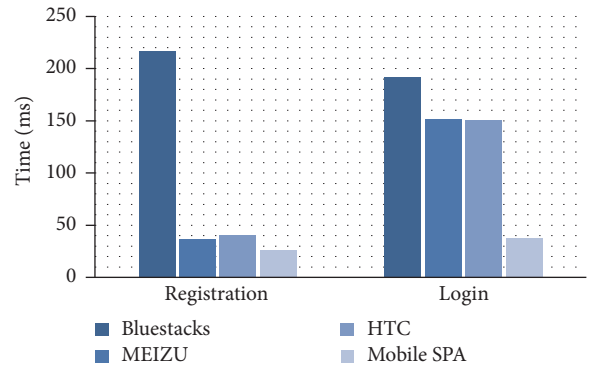


Figure 10: Comparison of experimental data in two phases of Case 4.

is the computation time on the server side, computing time on the mobile phone, and the transmission time. The total time of our scheme is 152ms and 151ms. Compared with the scheme of Mobile SPA, the time difference of our scheme is larger. However, our scheme has blind signature operation in the login phase and regeneration of the key operation and so on which greatly improve the security of the system. The time Users' using different Android device in registration and login is shown in Figure 10.

We can see from above test scenarios that the running time of our scheme during the registration phase is k1; the running time of our scheme in the login phase is t3+t2+t4; the total running time of our scheme is k1+t3+t2+t4 called w. The propagation delay time between the computer and the mobile phone is brief, so we consider t3 as the running time in the Android device. We can calculate the total running time w according to the Case 4. The comparison diagram of the time running in the distinct Android device is shown in Figure 11. We can see from Figure 11 that our scheme running in the Bluestacks costs the longest time. That is because there are some gaps in processor and set parameters between Bluestacks and Android mobile phone which influences the efficiency of our scheme.

In the storage capacity of mobile phones, the Mobile SPA scheme requires storing MAC key on the phone to authenticate with the CS. A server corresponds to a MAC key,

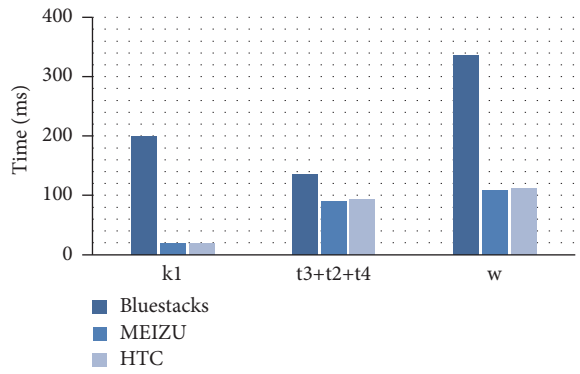


Figure 11: Comparison of the running time of different devices.

while the length of a MAC key is 128 bytes. In our scheme, the mobile device only needs to store the user's fingerprint data. The maximum length of fingerprint data is 512 bits (64 bytes). If the user registers 2000 servers, in the Mobile SPA scheme, the user requires 250kb storage capacity while the user requires 0.0625kb storage capacity in our FPPA scheme. Meanwhile, our scheme does not need to store data in the mobile phone for a long time.

In conclusion, although our scheme has a significant increase in the login phase, the user has little effect on the use in terms of user experience. Meanwhile, our scheme adds the blind signature operation and regeneration of the key operation and so on which greatly improve the security of the system. What is more, our scheme only stores the user's fingerprint in the mobile phone temporary. Because only a small amount information is stored in our scheme, the storage pressure of the mobile terminal is reduced. Therefore, our scheme has a high application value.

7. Summary

Passwords are widely deployed to secure users' access to cloud services in IIOT. However, passwords are prone to dictionary attack and phishing. Therefore, protecting the user's password when authenticating the user's identity has become a widespread concern. In this paper, we use the mobile phone to assist user's authentication. Specifically, we use both password and fingerprint to generate a secret key and then we use the secret key and HKDF function to generate the authentication key. Due to the characteristic of HKDF function, any adversary cannot get the key or password or decrypt the ciphertext through exhausting enumerating attacks. There is no user's secret information on the mobile phone, so even if the mobile phone is lost, it will not pose a security risk to the user's account. Finally, we prove the security of the scheme and evaluate the performance of our scheme. Results show that our scheme can resist dictionary attacks and attacks against the user's mobile phone. Meanwhile, there is no significant difference between our scheme and the existing three party authentication schemes, in terms of performance. In addition, our scheme reduces the storage pressure on the mobile phone. Overall, our scheme has high practicality.

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 work is supported by the National Key R&D Program of China (Grant no. 2017YFB0801805), the General Program of National Natural Science Foundation of China (Grant no. 61672415), the 111 Project (Grant no. B16037), and the open fund project of Science and Technology on Communication Networks Laboratory (Grant no. SXX18641X024).

References

[1] H. Xiong, H. Zhang, and J. Sun, "Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing," *IEEE Systems Journal*, pp. 1–22, 2018.

[2] C.-C. Lee, T.-H. Lin, and C.-S. Tsai, "A new authenticated group key agreement in a mobile environment," *Annals of Telecommunications-Annales des Télécommunications*, vol. 64, no. 11-12, pp. 735–744, 2009.

[3] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and Traceable Group Data Sharing in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 912–925, 2018.

[4] J. Xiong, Y. Zhang, and L. Lin, "ms-PoSW: A multi-server aided proof of shared ownership scheme for secure deduplication in cloud," *Concurrency & Computation Practice & Experience*, vol. 5, Article ID e4252, 2017.

[5] M. Zhang, Y. Yao, Y. Jiang, B. Li, and C. Tang, "Accountable mobile E-commerce scheme in intelligent cloud system transactions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1889–1899, 2018.

[6] D. Wu, F. Zhang, H. Wang, and R. Wang, "Security-oriented opportunistic data forwarding in Mobile Social Networks," *Future Generation Computer Systems*, vol. 87, pp. 803–815, 2018.

[7] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2958–2970, 2018.

[8] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1908–1920, 2017.

[9] N. Carlson, "Mark Zuckerberg broke into a facebook user's private email account," 2010, http://www.businessinsider.com.

[10] 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.

[11] C. Li, C. Lee, and C. Weng, "An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1133–1143, 2013.

[12] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[13] M. Wu, S. Garfinkel, and R. Miller, "Secure web authentication with mobile phones," in *Proceedings of the DIMACS workshop on usable privacy and security software*, 2004.

[14] D. Thanh, I. Jorstad, and T. Jonvik, "Strong authentication with mobile phone as security token," in *Proceedings of the IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pp. 777–782, IEEE, Macau, China, 2009.

[15] J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-is-believing: using camera phones for human-verifiable authentication," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 110–124, IEEE, May 2005.

[16] M. Mannan and P. C. van Oorschot, "Using a personal device to strengthen password authentication from an untrusted computer," in *Financial Cryptography and Data Security*, pp. 88–103, Springer Berlin Heidelberg, Berlin, Germany, 2007.

[17] C.-L. Chen, C.-C. Lee, and C.-Y. Hsu, "Mobile device integration of a fingerprint biometric remote authentication scheme," *International Journal of Communication Systems*, vol. 25, no. 5, pp. 585–597, 2012.

[18] T. Acar, M. Belenkiy, and A. Küpçü, "Single password authentication," *Computer Networks*, vol. 57, no. 13, pp. 2597–2614, 2013.

[19] Q. Su, J. Tian, and X. Chen, "A fingerprint authentication system based on mobile phone," in *Proceedings of the Audio and Video-Based Biometric Person Authentication*, pp. 151–159, Springer Berlin Heidelberg, New York, NY, USA, 2005.

[20] G. Starnberger, L. Froihofer, and K. M. Goeschka, "QR-TAN: Secure mobile transaction authentication," in *Proceedings of the International Conference on Availability, Reliability and Security (ARES '09)*, pp. 578–583, IEEE, Fukuoka, Japan, March 2009.

[21] Z. Matt, "FBI has accessed san bernardino shooter's phone without apple's help," 2016, http://gadgets.ndtv.com/mobiles/news.

[22] X. Boyen, "Hidden credential retrieval from a reusable password," in *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security (ASIACCS '09)*, pp. 228–238, ACM, New York, NY, USA, March 2009.

[23] X. Boyen, "Halting password puzzles," in *Proceedings of the 16th USENIX Security Symposium on USENIX Security Symposium*, pp. 119–134, ACM, Berkeley, Calif, USA, 2007.

[24] 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*, no. 99, pp. 2377–3782, 2018.

[25] 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, pp. 1–20, 2019.

[26] F. Li, Y. Han, and C. Jin, "Cost-effective and anonymous access control for wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 747–758, 2018.

[27] C.-C. Lee, M.-S. Hwang, and W.-P. Yang, "A new blind signature based on the discrete logarithm problem for untraceability," *Applied Mathematics and Computation*, vol. 164, no. 3, pp. 837–841, 2005.

[28] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments," *IEEE Systems Journal*, Article ID 2890126, pp. 1–11, 2018.

[29] S. Kumari, P. Chaudhary, C. Chen, and M. K. Khan, "Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications," *IEEE Access*, vol. 7, pp. 39717–39720, 2019.

[30] S. Kumari, X. Li, F. Wu, A. K. Das, K.-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.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 6045251, 11 pages
https://doi.org/10.1155/2019/6045251

Research Article

A Novel Device Identification Method Based on Passive Measurement

Wei Sun,[1] **Hao Zhang**,[2] **Li-jun Cai**,[2] **Ai-min Yu**,[2] **Jin-qiao Shi**,[3] **and Jian-guo Jiang**[2]

[1]*Beijing Jiaotong University, School of Computer and Information Technology, Beijing 100044, China*
[2]*Chinese Academy of Sciences, Institute of Information Engineering, Beijing 100093, China*
[3]*Beijing University of Posts and Telecommunications, School of Cyberspace Security, Beijing 100876, China*

Correspondence should be addressed to Wei Sun; 11112075@bjtu.edu.cn

Received 27 February 2019; Accepted 19 May 2019; Published 23 June 2019

Guest Editor: Fagen Li

Nowadays, with the continuous integration of production network and business network, more and more Industrial Internet of Things and Internal Office Network have been interconnected and evolved into a large-scale enterprise-level intraindustry network. Terminal devices are the basic units of internal network. Accurate identification of the type of device corresponding to the IP address and detailed description of the communication behavior of the device are of great significance for conducting network security risk assessment, hidden danger investigation, and threat warning. Traditional cyberspace surveying and mapping techniques take the form of active measurement, but they cannot be transplanted to large-scale intranet. Resources or specific targets in internal networks are often protected by firewalls, VPNs, gateways, and other technologies, so they are difficult to analyze and determine by active measurement. In this paper, a passive measurement method is proposed to identify and characterize devices in the network through real traffic data. Firstly, a new graph structure mining method is used to determine the server-like devices and host-like devices; then, the NAT-like devices are determined by quantitative analysis of traffic; finally, by qualitative analysis of the NAT-like device traffic, it is determined whether there are server-like devices behind the NAT-like device. This method will prove to be useful in identifying all kinds of devices in network data traffic, detecting unauthorized NAT-like devices and whether there are server-like devices behind the NAT-like devices.

1. Introduction

With the rapid development of information technology, the integration of production network and business network has become a reality. The interconnection of Industrial Internet of Things (IoT) and Internal Office Network has become a new networking trend, which not only greatly improves the production efficiency, but also achieves tight coupling of business work and production scheduling. A series of Stuxnet incidents that have erupted since 2010 are typical events that invade other networks from industrial IoT intrusions, which have had a huge negative impact on the global network, making losses difficult to measure. The methods of cyberattacks are more diverse, the devices targeted are more extensive, and the exploited vulnerabilities are more complicated, resulting in more serious threats. Specific security vulnerabilities often threaten a certain type of terminal device, which highlights the importance and urgency of fully understanding the distribution and attributes of cyberspace terminal devices.

Cyberspace surveying and mapping technology is an extension of network measurement, and network measurement technology is used for network mapping [1–3]. According to the measurement method, cyberspace surveying and mapping technology can be divided into active measurement and passive measurement [4–9]. For large-scale internal networks, especially those connected to industrial IoT, active measurement is not a good way. The main reasons are as follows: (1) Industrial IoT requires high real-time performance and stability and cannot tolerate the large number of data detection packages generated by active measurement. (2) Due to the limitation of the internal network transmission bandwidth, active measurement cannot be performed because it is easy to cause network congestion. (3) Active measurement methods are easily identified as attacks due to the placement of various security products in the internal network. (4) The

internal network has strict network boundaries and access control methods, so it is difficult to achieve reachability and coverage. In response to these active measurement problems, this paper presents a cyberspace surveying and mapping model guided by passive measurements. It can improve measurement results, reduce measurement complexity, and reduce network load.

In a large-scale industry internal network, hundreds of millions of traffic data are generated every day. Analyzing the data and mining their value are in line with the rules of passive measurement. The traffic data reflects the real state of the internal network and has a strong practical significance for comprehensively collecting the distribution and warning the potential threats of the devices in the cyberspace [10–13].

According to the devices corresponding to the IP addresses in the traffic data, we divide them into three categories: server-like devices, host-like devices, and NAT-like devices. Server-like devices include office application service devices, industrial production service devices, cloud computing devices, and data storage devices. Host-like devices include office computers and data collection terminals. NAT-like devices include firewalls, routers, gateways, and other address translation devices.

Normally, NAT devices [14] appearing in traffic data act as terminal devices, which makes devices hidden behind NAT-like devices arbitrarily access service resources in the internal network. Attackers may use devices hidden behind NAT-like devices to engage in illegal activities, such as launching an attack [15, 16], scanning the entire network, stealing data, maliciously spreading, or providing data services for violations, which can cause very serious problems. Therefore, it is necessary to periodically detect and filter out unauthorized NAT-like devices by analyzing data traffic.

In this paper, our goal is to identify the devices appearing in traffic data in order to achieve the surveying and mapping description of the internal network. We draw on the method of social relationship analysis and propose an unsupervised learning framework based on graph feature analysis and traffic analysis. Graph feature analysis is used to separate server-like devices and host-like devices, and traffic analysis is used to identify NAT-like devices. It should be noted that the communication relationship between the devices behind the NAT device does not appear in the traffic data, but these devices also belong to the device assets in the network. Therefore, we describe a set of validation analysis methods to identify server-like devices hidden behind NAT-like devices.

This paper makes the following research contributions:

(1) We use an unsupervised learning algorithm to classify network devices that appear in data traffic, which makes it feasible in internal networks compared to active measurements.

(2) We propose a graph feature analysis method for attribute mapping of all devices.

(3) We propose a description method, which is used to describe the attributes of the terminal devices.

(4) We propose a validation analysis method to determine whether there are server-like devices behind the NAT-like devices.

The model cannot only accurately identify the server-like devices and host-like devices, but also identify NAT-like devices with high accuracy. On this basis, a qualitative traffic analysis method is used to determine whether a server-like device exists behind the NAT-like device. By comparing the identified NAT-like devices with the asset list, an unauthorized NAT-like device can be found, which can eliminate network security risks, submitting devices' feature information to the security device and analyzing such information, which can improve the accuracy of the alarm.

2. Related Work

Zhao Fan et al. [17] describe the concept of cyberspace surveying and mapping. From the perspective of the Internet, it is summarized as the use of network detection, information collection, data processing, and data analysis to obtain physical resources and virtual resources. Through the positioning algorithm and the association analysis method, the physical resource is mapped to the geographic location, and the virtual resource is mapped to the simulated location. Finally, the detection results are drawn, which can intuitively reflect the state and development trend of cyberspace resources. Kohno et al. [18] propose a method for detecting network devices based on the offset value of the device clock. Fink [19] improves the method, introduces linear regression statistical method to judge the clock offset, and gives a calculation formula under certain degree of accuracy, so that the accuracy of such device recognition is controllable.

Wang Jianwei et al. [20], combining the degree of node and its neighbor node features, propose a method to evaluate the importance of the node with only local information, and the time performance is greatly improved. Kitsak et al. [21] propose Kl-shell in 2010, which is a method to determine the core of the network. The core idea of this method is to iteratively layer the nodes in the graph. The higher the number of layers, the more important the node is.

Gokcen et al. [22] generate 40 traffic attributes through the Net-Mate [23] tool and find the most effective eight attributes for identifying NAT devices using C4.5 [24] and Naive Bayes [25] machine learning algorithms. Another identification method in this paper uses payload information. However, in the internal network, the available traffic attributes are limited, and many attributes are not available due to confidentiality restrictions or encryption processing. Li Rui et al. [26] firstly express network traffic with 8-dimensional features, then filter the network traffic with an active value, and finally use the support vector machine to make two classes of NAT devices or host devices. Different servers store different resources. There may exist such a NAT-like device; the traffic generated with a particular server-like device is very high, but the level is not obvious in the total traffic analysis. Traversing server-like devices in turn, filtering out the NAT-like device connected to them by analyzing the traffic may achieve better classification results.

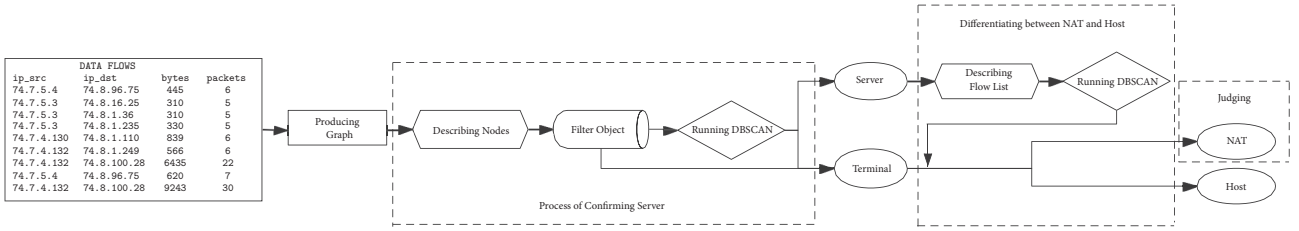


Figure 1: Model framework.

3. Recognition Method Based on Cluster Analysis

Figure 1 shows the framework of this method. The input we use is the traffic quad (source IP, destination IP, traffic size, number of packets), and each record represents a communication between the IP pair. The traffic data is converted into a graph structure; that is, the device IP address is used as a node, the communication relationship of the traffic data is used as an edge, and the information carried by the edge will be a triplet; the specific content will be described later. We will use the density-based clustering algorithm to classify devices twice. The whole process can be divided into three parts: One is to use explicit value to describe the importance of each node in the graph, then cluster analysis is used to identify server-like devices. The second is to obtain a list of terminal devices connected to the above server-like devices and use explicit numbers to describe the traffic level of the terminal device to server-like devices, thereby performing cluster analysis and separating the NAT-like devices and the host-like devices. Both of the above devices belong to the terminal device. The third is the analysis method of NAT-like device validation, to determine the existence of server-like devices behind. Next, we will discuss these three processes in detail. Finally, we briefly review the density-based clustering algorithm—DBSCAN.

3.1. Process of Confirming Server-Like Devices. Figure 2 is a simple example of node distribution, in which two central nodes represent two servers while the other nodes represent terminal devices, and the intersection part of the middle side indicates that these terminal devices communicate with both servers. We can classify nodes by characterizing node features.

Definition 1. Node degree refers to the number of edges associated with the node, also known as correlation degree.

The node degree of node i is formulized as

$$k_i = \left|\left\{e_{ij} \mid j \in V\right\}\right| \tag{1}$$

where V represents the node set of a graph and e_{ij} represents an edge between node i and node j.

Definition 2. The average degree of neighborhood refers to the average correlation degree of nodes in the neighborhood list of the node.

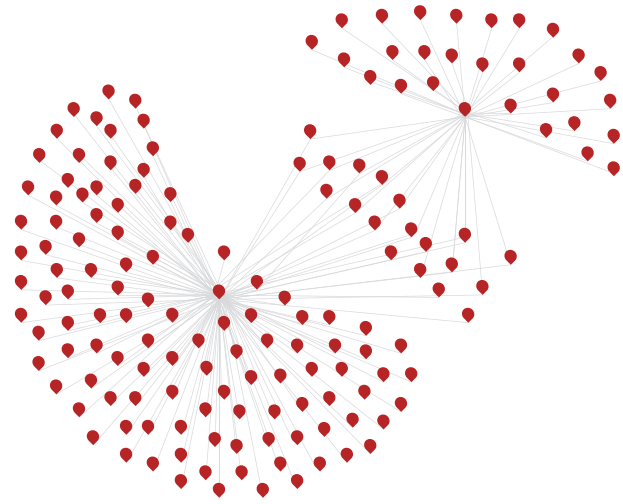


Figure 2: Graph structure example.

The average degree of neighborhood of node i is formulized as

$$k_{nn,i} = \frac{1}{|N(\mathrm{i})|} \sum_{j \in N(i)} k_j \tag{2}$$

where $N(i)$ represents the neighborhood of node i and k_j represents the correlation degree of node j which belongs to $N(i)$.

Definition 3. Traverse the nodes in the connected graph, all the paths of length 2 whose starting points are that nodes. Record the intermediate node of each path, and define the number of occurrences of each node as *the node fortress index.*

The node fortress index of node i is formulized as

$$f_i = \left|Mid_{\{path_{mn}=2 \mid m,n \in V\}}(\mathrm{i})\right| \tag{3}$$

where $\{path_{mn} = 2 \mid m, n \in V\}$ represents all paths of length 2 in the graph, and $Mid(i)$ indicates that the node i is exactly the intermediate node of the path.

Firstly, we calculate the average degree of neighborhood of each node, and then nodes are sorted from small to large. Stick out a mile; the average neighborhood of the server-like nodes is small. Secondly, there is a threshold α ($0< \alpha <1$), which means that the top α devices are selected to determine the server-like node. And the size of α depends on the experimental results. As a result, the node set which is used to

Input: G: Graph based on netflow; α: Select the front α section to filter nodes; *eps*: Neighbor radius to form a density area; *minPts*: Minimum number to form a high density area;
Output: Label of nodes: server or terminal;
$H \longleftarrow []$
$N \longleftarrow []$
for each *node* in G **do**
  $k_{node} \longleftarrow K(node)$ //calculate average neighbor degree of the node
  Put $(node, k_{node})$ in list H
$H \longleftarrow$sorting *node* according to k_{node} //ascending order
$N \longleftarrow \alpha$ of the front *node* in H //filtering nodes
$C \longleftarrow []$
for each n in N **do**
  $d_n \longleftarrow D(n)$ //calculate stronghold index of the node
  $f_n \longleftarrow F(n)$ //calculate degree of the node
  Put (d_n, f_n) in list C
Running DBSCAN with *eps* and *minPts* //with dataset C
return *outliers* //*outlier* represent Server, others represent Terminal

Algorithm 1: Process of server-like node detection.

identify the server-like nodes is reduced, and we will explain the details and prove that this step is necessary in later section. Thirdly, the two-dimensional features of all nodes in the node set are collected as the data to be clustered. As defined above, one of the characteristics is the correlation degree of each node, and the other is the fortress index of each node. Finally, we use DBSCAN algorithm for clustering analysis where outliers belong to server-like node and intracluster points belong to terminal node. Algorithm 1 shows the detailed process of identifying server-like node.

3.2. Differentiating between NAT-Like Devices and Host-Like Devices. The storage resources and uses vary between server-like devices, so the traffic levels of terminal devices for different server-like devices may also be different. At the same time, we assume that if two hosts connected to the same server-like device they may have similar traffic levels. Based on the situation we suggest above, for each server-like device, we find the list of terminal devices connected to it and then do clustering analysis. We believe that if a terminal device is a NAT-like device, it will connect with the corresponding server-like device more times, have more data packets, and generate more traffic. Therefore, we collect the three-dimensional features of all nodes in the terminal devices list as the data to be clustered. Then we use DBSCAN algorithm for clustering analysis where outliers belong to NAT-like devices and intracluster points belong to hosts. Imagine that if there are no NAT-like devices in a terminal devices list, all data will form a cluster without outliers. On the contrary, the NAT-like devices will become outliers and be marked. We repeat the above steps for each server-like node and finally intersect the results. Algorithms 2 gives the detailed process of distinguishing NAT-like devices from terminal devices.

3.3. Determining Whether Server-Like Devices Exist behind the NAT-Like Devices. According to the rules of capturing traffic data in intranet, the connection record between host and host is not included in the traffic data. If the connection record between a host-like device and a host-like device is found in the data flow, then it is certain that at least one of the communication parties is a NAT-like device, and the server-like device is hidden behind the NAT-like device.

We should also consider the connection records of server-like devices as source IP and NAT-like devices as destination IP. If the protocol used is TCP, then it is certain that the server-like device is connected to the server-like devices hidden behind the NAT-like device; if the protocol is UDP and the connection mode is query instead of answer, the server-like device sends query message to the server-like device behind the NAT-like device, proving that the server-like device is connected to the server-like device hidden behind the NAT-like device, as shown in Algorithms 3.

3.4. Density-Based Clustering Algorithm: DBSCAN. DBSCAN is an unsupervised machine learning algorithm, which assumes that classes can be determined by the compactness of the sample distribution; that is to say, the samples of the same class are closely linked.

DBSCAN algorithm has two parameters. One is the radius (Eps), which represents the circular neighborhood centered on fixed point P. The other parameter is the minimum number of points (MinPts) in the neighborhood centered on fixed point P. If there are at least MinPts in the neighborhood of Eps, the fixed point P is called the core point. If Q is located in the ϵ-neighborhood of P and this P is the core object, then Q is said to be directly density-reachable from P. For P and Q, if there is a sample sequence $p_1, p_2, \ldots, p_T$, satisfying $p_1 = P$ and $p_T = Q$, and if p_{t+1} is directly density-reachable from p_t, then P is said to be density-reachable from Q. That is to say, the density-reachable relation satisfies the transmissibility. For P and Q, if there is a core point m, so that

Input: G: Graph based on netflow; S: List of Serve Detected from Algorithm 1; T: List of Terminal Detected from Algorithm 1; *eps*: Neighbor radius to form a density area; *minPts*: Minimum number to form a high density area;
Output: Label of nodes: NAT or host;

```
N ⟵ []
for each s in S do  //ergodic per server
    L ⟵ []
    for each t in T do  //ergodic per terminal
        if from s to t exist edge in G then  //G belongs to undirected graph
            Put t in list L  //creat special terminal list about this server
    C ⟵ []
    for each l in L do
        t_l ⟵ T(l)  //time of communication between s and l
        p_l ⟵ P(l)  //package's number between s and l
        f_l ⟵ F(l)  //total flow data between sand l
        Put (t_l, p_l, f_l) in list C
    O ⟵ []
    Running DBSCAN with eps and minPts  //with dataset C
    O ⟵ outliers  //collecting outliers with list C
    N ⟵ N ∪ O  //integer O found by servers
return N  //N represent NAT, others represent Host
```

Algorithm 2: Differ NAT-like node and host-like node.

Input: S: List of Serve Detected from Algorithm 1; N: List of NAT Detected from Algorithm 2; H: List of Host Detected from Algorithm 2; R: List of Data Flow Recording;
Output: List of NATs with Servers behind them;

```
O ⟵ []
for each r in R do  //traverse through each record
    for each n in N do  //traverse through each NAT
        if n is src of r and dst is in H then  // T -> T
            Put src in list O
        if n is dst of r and src is in S then  // S -> T
            if protocol is HTTP then
                Put dst in list O
            if protocol is UDP and pattern is query then  //query or answer
                Put dst in list O
return O  //N represent NATs with Servers behind them
```

Algorithm 3: Process of judging servers behind NATs.

both P and Q can be density-reachable from the core point m, that P and Q are density-connected.

Steps to run DBSCAN:

(a) We travel through each point to find all the core points.

(b) Starting from a core point, we expand to a region with density-reachable relation and obtain a region that contains the core point and boundary points, in which any two points are density-connected.

The density-connected samples are grouped into the same class; in this way we can get the clustering results, as shown in Figure 3.

4. Evaluation

In order to test the effectiveness of the above method in device identification in the internal network, we will describe the following four aspects.

4.1. Dataset. Our dataset comes from Elasticsearch and the data collection process is shown in Figure 4. First, the traffic information is collected into the Traffic Collection Server through the mirroring interface; then, through the inspection and procedure of the PTD software, the traffic information is transmitted to the NSQ, a distributed real-time messaging platform; finally, Logstash acts as an intermediate station and the information is copied to the Elasticsearch, a Lucene-based search server. We can get traffic records for a certain

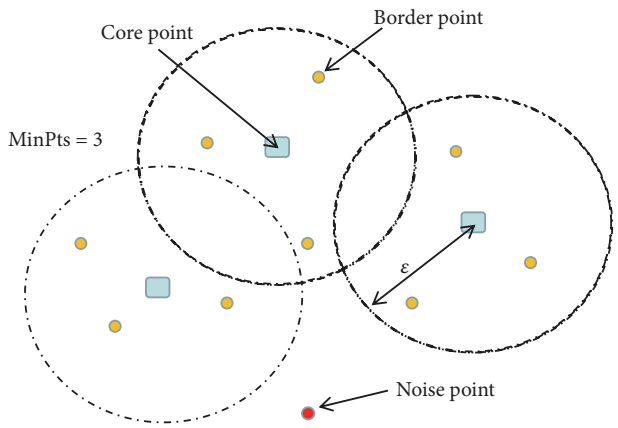


Figure 3: DBSCAN.

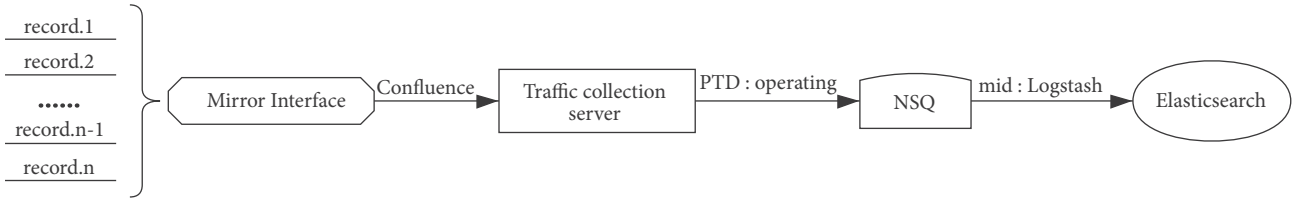


Figure 4: Process of collecting.

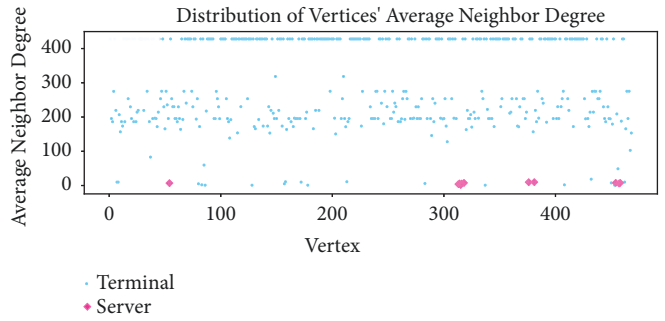


Figure 5: Line chart of the neighborhood average distribution.

day from Elasticsearch. After simple data processing, each traffic record is a four-tuple (src-ip, dst-ip, bytes, packets) that represents a communication connection between the source IP and the destination IP. In traffic data, the number of IPs is 468, and the number of records is 2,634,182. If it converted to a graph, the graph will contain 468 nodes and 1603 edges. The device asset list can display the device type corresponding to the IP address, which can be regarded as a tag set.

4.2. Selected Attributes. In this section, we discuss the two classification processes separately.

(1) Process of Confirming Server-Like Devices. There are many metrics for the importance of the nodes in graph, such as Degree Centrality, Betweenness Centrality, and Closeness Centrality. However, a single indicator only characterizes the structural features of the graph partly. In order to more accurately and comprehensively characterize the features of the network, multiple indicators are needed to reflect the features of the network. Combining theoretical research and statistical analysis, we summarize the measurement indicators of various characterizations. The neighborhood average can reduce the set of nodes to be classified. The value of the neighborhood average of the server-like nodes will always be small. On the contrary, the value of the neighborhood of the terminal nodes are almost larger, as shown in Figure 5.

We find that the association between node correlation and node fortress index is very large. The most important point is that these two indicators can be divided into two classes: server-like node and terminal node, as shown in Figure 6.

We map the above two metrics as an array into a two-dimensional space. Under the premise of adjusting the two parameters of the DBSCAN algorithm (Eps and MinPts), the two types of data points are separable. Due to the data overlap, the number of data points in the marked area in Figure 7 is 26.

(2) Differentiating between NAT-Like Devices and Host-Like Devices. As mentioned above, we believe that if a terminal

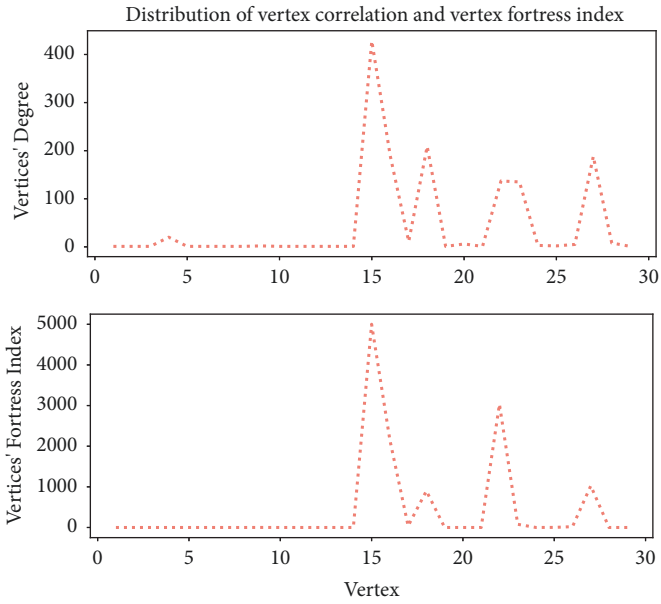


Figure 6: Distribution of node correlation and node fortress index.

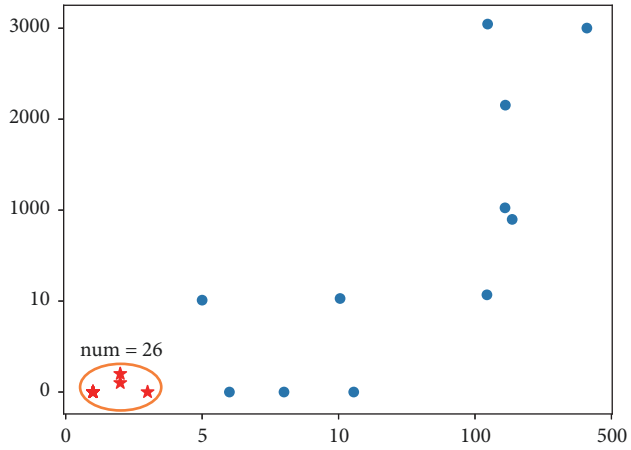


Figure 7: Distribution of data in two dimensions.

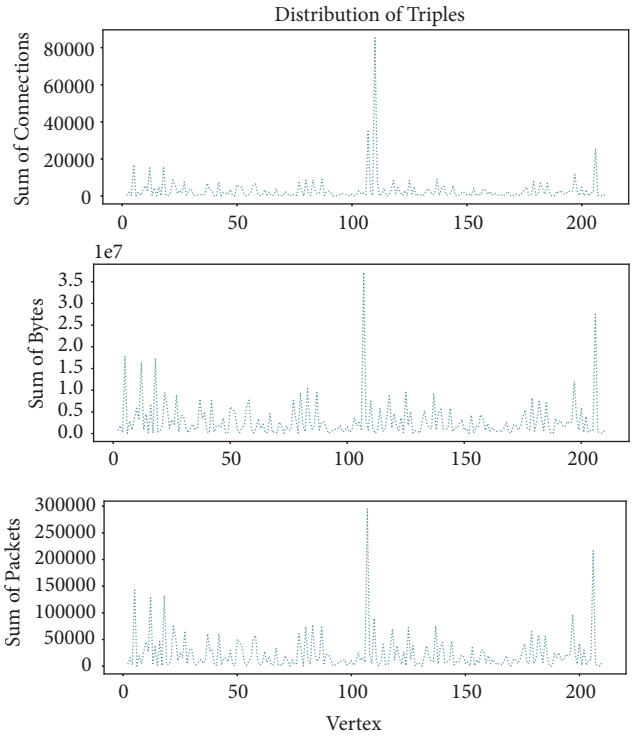


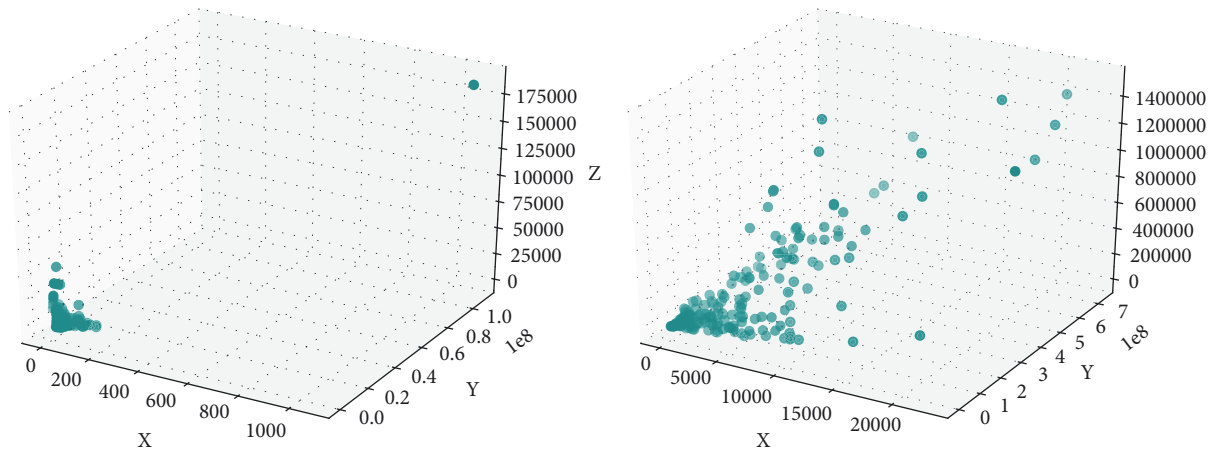
Figure 8: Distribution of triples.

device is actually a NAT-like device, it will have more connections with the corresponding server-like device. Then it will receive or send more packets as well as generating more traffic information. We have identified the server-like devices in the internal network. Then, for a server-like node, the distribution of the three indicators of the terminal devices list is shown in Figure 8, which indicates that the three indicators are roughly the same for the terminal device.

First of all, for each server-like node, we map its corresponding triplet data into 3D space. Then, we use DBSCAN algorithm for cluster analysis. In order to better demonstrate the adaptability of DBSCAN algorithm to this problem, we select two server-like nodes. One of the two does have connection with a NAT-like device, as shown in Figure 9(a). The other does not have connection with any NAT-like devices, as shown in Figure 9(b). It is obvious that there is no outlier in Figure 9(b).

4.3. Parameter Determination. In this section, we will discuss three classification processes and qualitative analysis process, respectively.

(1) Process of Confirming Server-Like Devices. In the process of confirming server-like devices, the parameter MinPts is not sensitive to clustering results, so we set the parameter MinPts value to 4 in all tests. As shown in Figure 10, with



(a) Distribution with NAT-like devices

(b) Distribution without NAT-like devices

Figure 9

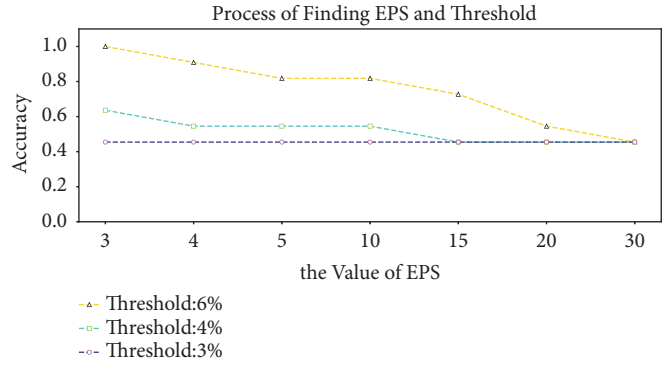


Figure 10: Accuracy of identifying server-like node.

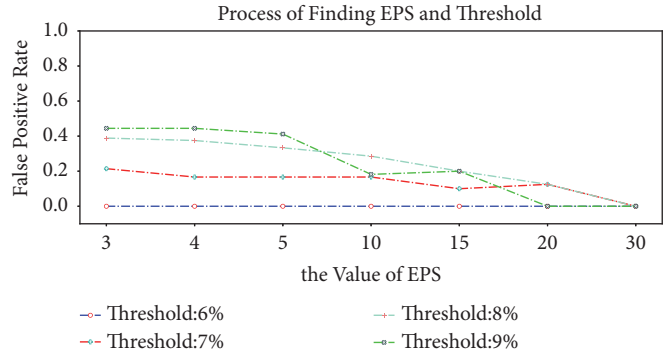


Figure 11: False rate of identifying server-like node.

the increase of Eps value, the accuracy of server-like devices recognition gradually decreases. At the same time, the minimum threshold is 0.06; otherwise, it will affect the accuracy of recognition results. When the Eps value is 3 and the threshold value is 0.06, the recognition accuracy of server-like devices is the highest. As shown in Figure 11, with the increase of Eps, the false positive rate of recognition decreases, but the recognition accuracy decreases significantly at this time. We also find that increasing the threshold will lead to greater false alarm rate. When the threshold is 0.06, the false alarm rate of the model is 0. After many tests, when Eps value is 3, MinPts value is 4, and threshold value is 0.06, the recognition effect is the best.

(2) Differentiating between NAT-Like Devices and Host-Like Devices. As shown in Figure 12, the recall rate increases with the increase of Eps value. When the Eps value is 0.7-0.9 and the MinPts value is 5, the recall rate is higher and stable. As shown in Figure 13, the accuracy decreases with the increase of Eps value. When Eps value is 0.7 and MinPts

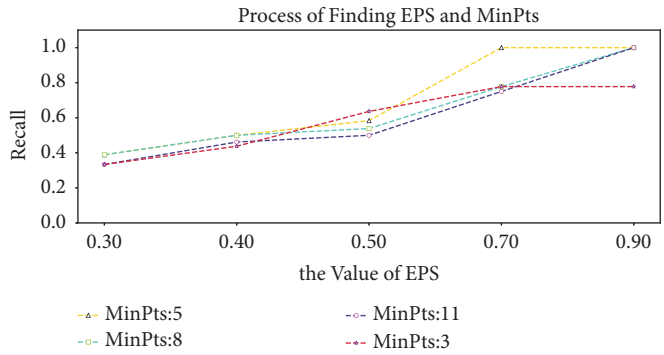


Figure 12: Recall rate of NAT-like devices.

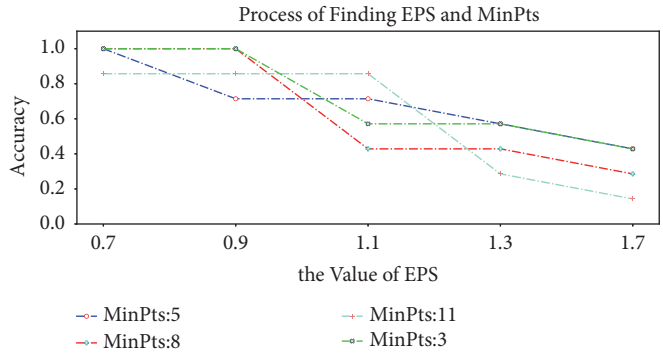


Figure 13: Accuracy of identifying NAT-like devices.

value is 3-5, the accuracy rate is higher and stable. We need to find parameter combinations with high recall and high accuracy. After many tests, when the Eps value is 0.6-0.8 and the MinPts value is 5, the recognition effect is the best.

(3) Determining Whether Server-Like Devices Exist behind a NAT-Like Device. We need to consider two situations where there are server-like devices behind NAT-like devices: one is that the server-like device only serves the devices behind the NAT-like device, so the traffic data will not have corresponding connection records; the other is that the service objects of the server-like device are the whole network devices, which can be accessed by the devices in the intranet, and then the traffic data information will have corresponding connection records. Our validation analysis process is only used for the latter situation. In the process of analysis, we find 4 such connection records, among which the source IP is a host-like device in the intranet and the destination IP is the NAT-like device. Therefore, we can conclude that there is a server-like device behind the NAT-like device.

4.4. Intranet Visualization. As mentioned above, the cyberspace surveying and mapping technology ultimately draws the detection result. We present a visual result of a dataset, as shown in Figure 14. It should be noted that the visualization part is cropped. NAT1 represents a NAT-like device with no server-like device behind it, and NAT2 represents a NAT-like device with a few server-like devices behind it.

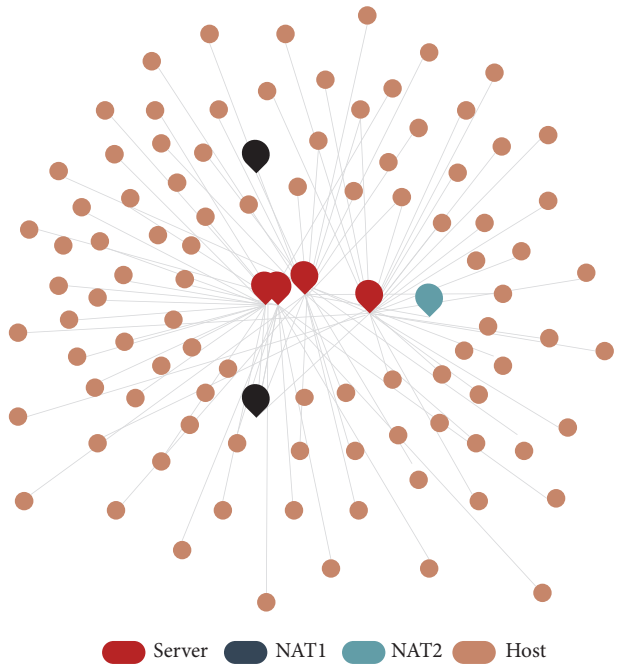


Figure 14: Visualization of the internal network.

5. Conclusion

Based on the unsupervised clustering algorithm—DBSCAN—this paper identifies the devices in the internal

network by a passive measurement. This process can be divided into two processes. One is to use the graph structure features to identify the server-like devices and terminal devices in the internal network. In this process, we use a filtering method, which can prevent the highly associated terminal devices from being misidentified as server-like devices. The second step is using traffic analysis method to divide terminal devices into NAT-like devices and host-like devices. During this process, a method of traversing the server-like devices to detect the NAT-like devices is adopted, which can make the classification result better. The framework is effective. It can identify network devices existing in the data traffic information and detect the existence of unauthorized NAT-like devices. The surveying and mapping information obtained by this framework provides important data support for improving the effectiveness and intelligence of analysis methods such as causal association, attack scene correlation, and subject-object association. In the data traffic information, only the related traffic of these three types of devices appears, and other devices such as switches and hubs do not appear in Elasticsearch. We will continue to expand our research to achieve more comprehensive cyberspace surveying and mapping.

Data Availability

The data that support the findings of this study are not publicly available due to restrictions as the data contain sensitive information about a real-world intraindustry network. Access of the dataset is restricted by the original owner. People who want to access the data should send a request to the corresponding author, Wei Sun, who will apply for permission of sharing the data from the original owner.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Motamedi, R. Rejaie, and W. Willinger, "A survey of techniques for internet topology discovery," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1044–1065, 2015.

[2] A. Dainotti, K. Benson, A. King, M. Kallitsis, and Glatz. E., "Errata for: Estimating internet address space usage through passive measurements (SIGCOMM CCR (Vol. 44, Issue 1, January, 2014)," *Acm Sigcomm Computer Communication Review*, vol. 44, no. 2, pp. 99-100, 2014.

[3] K. Levchenko, A. Dhamdhere, B. Huffaker, K. Claffy, M. Allman, and V. Paxson, "PacketLab: A universal measurement endpoint interface," in *Proceedings of the 2017 ACM Internet Measurement Conference, IMC 2017*, pp. 254–260, ACM, November 2017.

[4] W. Sun, J. Jiang, and M. Su, "A passive-measurement-guided tree network surveying and mapping model," in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 646–651, Guangzhou, China, June 2018.

[5] M. Li, Y. Sun, Y. Jiang, and Z. Tian, "Answering the min-cost quality-aware query on multi-sources in sensor-cloud systems," *Sensors*, vol. 18, no. 12, p. 4486, 2018.

[6] W. Han, Z. Tian, Z. Huang, S. Li, and Y. Jia, "Bidirectional self-adaptive resampling in internet of things big data learning," *Multimedia Tools and Applications*, 2018.

[7] Z. Wang, C. Liu, J. Qiu, Z. Tian, X. Cui, and S. Su, "Automatically traceback rdp-based targeted ransomware attacks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7943586, 13 pages, 2018.

[8] Z. Tian, S. Su, W. Shi, X. Du, M. Guizani, and X. Yu, "A data-driven method for future Internet route decision modeling," *Future Generation Computer Systems*, vol. 95, pp. 212–220, 2019.

[9] J. Qiu, Y. Chai, Y. Liu, Z. Gu, S. Li, and Z. Tian, "Automatic non-taxonomic relation extraction from big data in smart city," *IEEE Access*, vol. 6, pp. 74854–74864, 2018.

[10] Y. Wang, Z. Tian, H. Zhang, S. Su, and W. Shi, "A privacy preserving scheme for nearest neighbor query," *Sensors*, vol. 18, no. 8, p. 2440, 2018.

[11] Z. Tian, Y. Cui, L. An et al., "A real-time correlation of host-level events in cyber range service for smart campus," *IEEE Access*, vol. 6, pp. 35355–35364, 2018.

[12] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. H. Tian, "Towards a comprehensive insight into the eclipse attacks of tor hidden services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1584–1593, 2018.

[13] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, "Trust architecture and reputation evaluation for internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, pp. 1–9, 2018.

[14] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC Editor RFC2663, 1999.

[15] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets," in *Proceedings of the ACM SIGCOMM 2001- Applications, Technologies, Architectures, and Protocols for Computers Communications-*, pp. 15–26, USA, August 2001.

[16] X. Yu, Z. Tian, J. Qiu, and F. Jiang, "A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5823439, 11 pages, 2018.

[17] F. Zhao, X.-y. Luo, and F.-l. Liu, "Research on cyberspace surveying and mapping technology," *Chinese Journal of Network and Information Security*, vol. 9, no. 2, pp. 1–11, 2016.

[18] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[19] R. Fink, "A statistical approach to remote physical device fingerprinting," in *Proceedings of the Military Communications Conference, MILCOM 2007*, USA, October 2007.

[20] J.-W. Wang, L.-L. Rong, and T.-Z. Guo, "A new measure method of network node importance based on local characteristics," *Journal of Dalian University of Technology*, vol. 50, no. 5, pp. 822–826, 2010.

[21] M. Kitsak, L. K. Gallos, S. Havlin et al., "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.

[22] Y. Gokcen, V. A. Foroushani, and A. N. Z. Heywood, "Can we identify NAT behavior by analyzing traffic flows?" in *Proceedings of the 2014 IEEE Computer Society's Security and Privacy Workshops, SPW 2014*, pp. 132–139, USA, May 2014.

[23] A. Dupuy, S. Sengupta, O. Wolfson, and Y. Yemini, "NETMATE: a network management environment," *IEEE Network*, vol. 5, no. 2, pp. 35–40, 1991.

[24] Quinlan J R. C4.5: programs for machine learning, 1992.

[25] J.-H. Xue and D. M. Titterington, "Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes"," *Neural Processing Letters*, vol. 28, no. 3, pp. 169–187, 2008.

[26] R. Li, H. Zhu, Y. Xin et al., "Remote NAT detect algorithm based on support vector machine," in *Proceedings of the International Conference on Information Engineering & Computer Science*, IEEE, 2009.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 2062697, 12 pages
https://doi.org/10.1155/2019/2062697

Research Article

Security Cryptanalysis of NUX for the Internet of Things

Yu Liu[1], **Xiaolei Liu**[2] **and Yanmin Zhao**[3]

[1]*School of Computer Engineering, Weifang University, Weifang 261061, China*
[2]*Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China*
[3]*Department of Computer Science, Faculty of Engineering, The University of Hong Kong, 999077, Hong Kong*

Correspondence should be addressed to Yu Liu; yuliu@wfu.edu.cn and Xiaolei Liu; lxlmath@mail.sdu.edu.cn

Received 15 November 2018; Revised 13 May 2019; Accepted 23 May 2019; Published 12 June 2019

Guest Editor: Fagen Li

In order to adopt the restricted environment, such as radio frequency identification technology or sensor networking, which are the important components of the Internet of Things, lightweight block ciphers are designed. NUX is a 31-round iterative ultralightweight cipher proposed by Bansod *et al*. In this paper, we examine the resistance of NUX to differential and linear analysis and search for $1 \sim 31$-round differential characteristics and linear approximations. In design specification, authors claimed that 25-round NUX is resistant to differential and linear attack. However, we can successfully perform 29-round differential attack on NUX with the 22-round differential characteristic found in this paper, which is 4 rounds more than the limitation given by authors. Furthermore, we present the key recovery attack on 22-round NUX using a 19-round linear approximation determined in this paper. Besides, distinguishing attack, whose distinguisher is built utilizing the property of differential propagation through NUX, is implemented on full NUX with data complexity 8.

1. Introduction

The Internet of Things is defined as a variety of devices and technologies such as sensors, radio frequency identification (RFID) technology, global positioning systems, infrared sensors, laser scanners, and gas sensors. Its essence is to use RFID technology to realize the automatic identification of items, the interconnection, and sharing of information through the computer Internet. In this kind of new cryptography environment, RFID technology and sensor networking have similar properties, such as weak computation abilities, small storage spaces, and strict power constraints. Therefore, traditional block ciphers are not suitable for this kind of extremely constrained environment. Hence, lightweight block ciphers are put forward for restricted environment and have shown importance in various applications. Recently, copious lightweight block ciphers are designed to maintain security under limited resource conditions, such as PRESENT [1], LBlock [2], and PRINCE [3]. And many cryptographers are concerned about the security of lightweight block ciphers.

Differential analysis, which is a chosen-plaintext attack, is proposed by Biham and Shamir to analyze DES [4]. Differential analysis studies probability propagation property in the encryption/decryption process. This method seeks for high probability differential characteristics to perform a distinguishing attack or a key-recovery attack. Later, linear analysis, which is the duality of differential analysis and is a known-plaintext attack, is presented by Matsui in EUROCRYPT'93 [5]. Similarly, linear analysis studies the linear relationship between plaintexts and ciphertexts and finds linear approximations with high probability to build a distinguisher or carry out a key-recovery attack. In addition, these two methods are the most popular cryptanalysis methods nowadays. They have been used to analyze many ciphers and should be taken into consideration for designing a new cipher scheme [6–11].

NUX is a 31-round iterative lightweight block cipher proposed by Bansod *et al.* [12] which adopts generalized Feistel structure. And it supports 64-bit blocks and 80/128-bit keys. For the version with 128-bit key, the author pointed out that NUX needs 1022 GEs, which is less than all existing ciphers.

Table 1: Comparison of tails on NUX.

Method	Rounds	Probability/bias	Reference
Differential	25	2^{-90}	[12]
	25	2^{-71}	Section 3.1
	31	2^{-88}	Section 3.1
Linear	25	2^{-66}	[12]
	25	2^{-41}	Section 4.1
	31	2^{-50}	Section 4.1

Table 2: Summary of attacks on NUX.

Attack type	Rounds	Time	Date	Memory (Bytes)	Reference
Differential	25	-	2^{90}	-	[12]
	29	$2^{121.73}$	2^{61}	2^{96}	Section 3.2
Linear	25	-	2^{132}	-	[12]
	25	2^{126}	$2^{63.7}$	2^{70}	Section 4.2
Distinguishing	31	8	8	0	Section 5

In terms of security, they expected that NUX could resist differential/linear analysis. They examined the resistance of NUX against differential/linear cryptanalysis. At last, they showed that NUX cipher with no less than 25 round is enough to resist linear or differential attack. Besides, a biclique attack on NUX is presented in [12].

Our Contribution

(i) 1~31-round differential characteristics of NUX are searched for, and 10 25-round differential characteristics are found with probability 2^{-71}, which are better than the one with probability limitation 2^{-90} given in the design script, shown in Table 1.

(ii) The resistance to the linear analysis for $1 \sim 31$-round NUX is examined, and 48 25-round linear approximations are presented with absolute value of bias to be 2^{-41} better than the limitation on bias presented in the design script, which is 2^{-66}, depicted in Table 1.

(iii) For full NUX, the probability of the best differential characteristic is 2^{-88}, and the absolute value of bias for the best linear approximation is 2^{-50}, which are described in Table 1.

(iv) Using 22-round differential characteristic with probability 2^{-58} obtained in this paper, 29-round differential attack is performed with time, data, and memory complexity to be $2^{121.73}$ 29-round encryptions and 2^{61} and 2^{96} bytes, respectively. Furthermore, based on a 19-round linear approximation with bias to be 2^{-30}, 25-round linear attack is executed with time complexity 2^{126} 25-round encryptions, data complexity $2^{63.7}$, and memory complexity 2^{70} bytes. Till now, these two attacks are the best differential attack and best linear attack, respectively. A summary of our attacks is given in Table 2.

(v) Utilizing the property of difference propagation through NUX, distinguishing attack can be implemented on full NUX with data complexity 8, which is depicted in Table 2.

The organization of the paper is as follows. The notations and description of NUX are given in Section 2. Section 3 shows the differential characteristics and differential attack on 29-round NUX. In Section 4, the linear approximations and linear attack on 25-round NUX are introduced. Section 5 describes distinguishing attacks between full NUX and random permutations. Finally, Section 6 concludes the paper.

2. Preliminaries

This section will list notations and operations used in this paper and describe NUX.

2.1. Notations and Operations

(i) $\{0,1\}^n$: The set of strings with n bits length.

(ii) $x[i]$: The i-th bit of x, if $x \in \{0,1\}^n$.

(iii) $x[i:j]$: The j-th bit to i-th bit of x, if $x \in \{0,1\}^n$.

(iv) $x[i,j]$: The i-th and j-th bits of x, if $x \in \{0,1\}^n$.

(v) $x \parallel y$: Concatenation of x and y, if $x, y \in \{0,1\}^n$.

(vi) $\ggg$: Right cyclic shift operation.

(vii) $\lll$: Left cyclic shift operation.

(viii) $\oplus$: XOR operation.

2.2. Description of NUX. NUX is a 31-round ultralightweight cipher based on generalized Feistel network. It supports a key length of 128/80 bits and a block length of 64 bits. The round function is illustrated in Figure 1.

There are two F-functions F_1 and F_2, which are constructed by four 4×4 S-boxes and a circular shift operator.

Table 3: S-box used in NUX.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S(x)	e	7	8	4	1	9	2	f	5	a	b	0	6	c	d	3

Table 4: Bit permutation table P in NUX.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	15	11	7	3	2	14	10	6	5	1	13	9	8	4	0	12

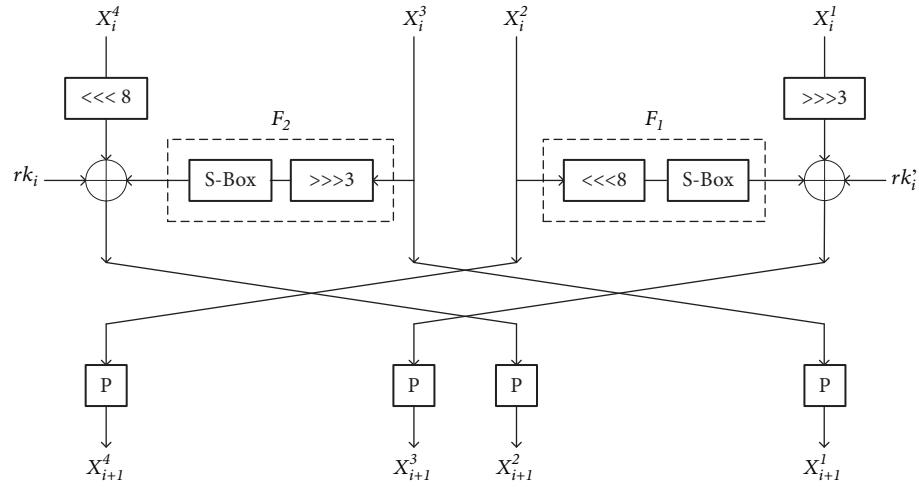


Figure 1: The round function of NUX.

The S-box is represented in Table 3. Then F_1 and F_2 can be depicted as follows:

$$\begin{aligned} F_1(x) &= S(x[7:4]) \parallel S(x[3:0]) \parallel S(x[15:12]) \parallel \\ &\quad S(x[11:8]), \\ F_2(x) &= S(x[2:0] \parallel x[15]) \parallel S(x[14:11]) \parallel \\ &\quad S(x[10:7]) \parallel S(x[6:3]). \end{aligned} \tag{1}$$

The 64-bit input of the i-th round, X_i, is divided into 4 blocks of 16 bits, named X_i^4, X_i^3, X_i^2, and X_i^1, respectively.

$$X_i = X_i^4 \parallel X_i^3 \parallel X_i^2 \parallel X_i^1, \tag{2}$$

and then there are the following formulas:

$$\begin{aligned} X_{i+1}^4 &= P\left(X_i^2\right), \\ X_{i+1}^3 &= P\left(\left(X_i^1 \ggg 3\right) \oplus F_1\left(X_i^2\right) \oplus rk_i'\right), \\ X_{i+1}^2 &= P\left(\left(X_i^4 \lll 8\right) \oplus F_2\left(X_i^3\right) \oplus rk_i\right), \\ X_{i+1}^1 &= P\left(X_i^3\right), \end{aligned} \tag{3}$$

where P is a 16-bit permutation, which is depicted in Table 4.

After 31 rounds, the ciphertext will be acquired as $X_{31}^4 \parallel X_{31}^3 \parallel X_{31}^2 \parallel X_{31}^1$. The key schedule is omitted and interested readers are referred to [12] for more details about NUX.

3. Differential Attack on 29-Round NUX

In this section, how to search for differential characteristics of NUX will be described. And then a key-recovery attack is conducted on 29-round NUX.

3.1. Differential Characteristics of NUX. To search for the differential characteristic of NUX, the different propagation between round functions should be considered. And how differences propagate through S-boxes should also be taken into account. When a difference passes through an S-box, the output difference and probability are determined by looking up the XOR difference distribution table (DDT) of the S-box.

Algorithm 1 is designed to search for differential characteristics of NUX, and notations used in this algorithm are depicted in Figure 2. From the structure of NUX, it is obvious that the probability of one round can be 1; that is, all the 8 S-boxes are passive. So a special $n-1$-round differential characteristic, $(\Delta_0^3 = \Delta_0^2 = 0)$, is chosen, and one-round differential characteristic with probability 1 is added to catch n-round differential characteristics. Meanwhile, Δ_i^4 and Δ_i^3, which lie in the two branches on the left side, will not affect Δ_i^2 and Δ_i^1, the two branches on the right side, during propagation in NUX. So differences of two branches on the left or right are set to 0, and the difference propagation in the other two branches will be focused on during the encryption process. In this way, the number of active S-boxes is always no

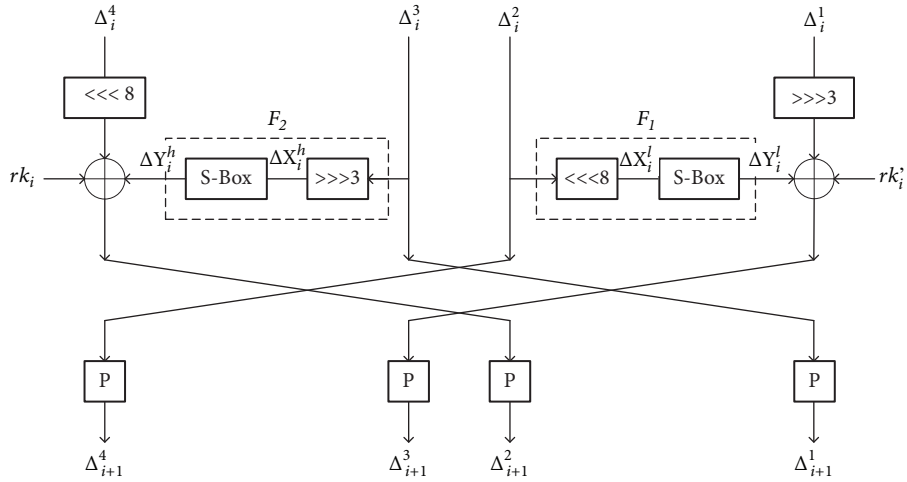


Figure 2: Differential representation of NUX.

Input: S-box S, Probability threshold $PrLim[30]$.
Output: A differential trail with the best probability.
(1) Generate the DDT of S-box.
(2) Store all 2, 4 and the corresponding input/output differences in DDT in the table DDT_P
(3) **for** Each of 8 S-boxes in the first round **do**
(4) **for** all non-zero entities in DDT **do**
(5) **if** S-box is in F_2 **then**
(6) $\Delta_1^3 = \Delta X_1^h \lll 3, \Delta_1^4 = \Delta_1^2 = \Delta_1^1 = 0$.
(7) Store them in $R_{in}[1]$.
(8) **else**
(9) $\Delta_1^2 = \Delta X_1^l \ggg 8, \Delta_1^4 = \Delta_1^3 = \Delta_1^1 = 0$.
(10) Store them in $R_{in}[1]$.
(11) Calculate $\Delta_0^4, \Delta_0^3, \Delta_0^2, \Delta_0^1$, and store them in $R_{in}[0]$.
(12) Calculate $\Delta_2^4, \Delta_2^3, \Delta_2^2, \Delta_2^1$, and store them in $R_{in}[2]$.
(13) The probability is recorded as $Pr[1]$.
(14) **for** $i \longleftarrow 2$ to 30 **do**
(15) **if** $\Delta_i^4 = \Delta_i^3 = 0$ **then**
(16) $\Delta X_i^l = \Delta_i^2 \lll 8$
(17) Travel DDT_P to get ΔY_i^l and its corresponding probability $PrRnd$
(18) **else if** $\Delta_i^2 = \Delta_i^1 = 0$ **then**
(19) $\Delta X_i^h = \Delta_i^3 \ggg 3$
(20) Travel DDT_P to get ΔY_i^h and the corresponding probability $PrRnd$
(21) Keep $Pr[i-1] \times PrRnd$, which greater than threshold $PrLim[i]$
(22) Calculate $\Delta_{i+1}^4, \Delta_{i+1}^3, \Delta_{i+1}^2, \Delta_{i+1}^1$, and store them in $R_{in}[i+1]$.
(23) $Pr[i] \Longleftarrow Pr[i-1] \times PrRnd$
(24) **for** Each of 8 S-boxes in the first round **do**
(25) **for** all non-zero entities in DDT **do**
(26) Find the largest $Pr[i]$ and store it in $PrMax[i]$
(27) Output different characteristics with probability $PrMax[i]$

Algorithm 1: Algorithm for differential characteristic on NUX.

more than 4 in one round, which can make the probability of the differential characteristic as large as possible. The results of our search algorithm are listed in Table 5.

Furthermore, the minimal numbers of active S-boxes of 1~5-round differential characteristics are shown in the design manuscript [12]. So the minimal number of active S-boxes is also studied, and the minimum active S-boxes of 4 5-round differential characteristics are less than the ones given in [12], which are presented in Table 6.

3.2. Differential Attack on NUX. A 22-round differential characteristic is chosen with probability to be 2^{-58}, which is shown in Table 7, and 7 rounds are extended forward. The description is given in Figure 3. There are $\Delta_0^4 = 0x0400$,

Table 5: Differential characteristics of NUX.

Rounds	Probability	Number of tails
2	2^{-2}	192
3	2^{-4}	56
4	2^{-6}	4
5	2^{-7}	1
6	2^{-9}	2
7	2^{-11}	2
8	2^{-15}	8
9	2^{-15}	1
10	2^{-20}	16
11	2^{-24}	24
12	2^{-28}	4
13	2^{-31}	8
14	2^{-37}	4
15	2^{-41}	18
16	2^{-45}	2
17	2^{-46}	1
18	2^{-49}	4
19	2^{-52}	4
20	2^{-53}	2
21	2^{-58}	48
22	2^{-58}	2
23	2^{-63}	32
24	2^{-67}	49
25	2^{-71}	10
26	2^{-73}	4
27	2^{-75}	4
28	2^{-79}	16
29	2^{-79}	2
30	2^{-84}	32
31	2^{-88}	48

Table 6: Minimal number of active S-boxes from differential characteristic.

Reference	Number of active S-boxes				
	Rounds				
	1	2	3	4	5
[12]	0	1	2	5	9
Section 3.1	0	1	2	3	3

$\Delta_{22}^3 = 0x2080$, and $\Delta_0^3 = \Delta_0^2 = \Delta_0^1 = \Delta_{22}^4 = \Delta_{22}^2 = \Delta_{22}^1 = 0x0000$ according to the 22-round differential distinguisher. Then $\Delta_{23}^4 = \Delta_{23}^3 = 0x0000$, $\Delta_{23}^1 = 0x0050$, and Δ_{23}^2 satisfy the form as $(00 * * * 00 * * * 000 * *0)$, where $* \in \{0, 1\}$ can be deduced. And the output difference of the 28-th round is $\Delta_{29}^4 = \Delta_{29}^3 = 0x0000$. X_{23}^2, $X_{23}^{2'}$, X_{23}^1, and $X_{23}^{1'}$ can be gotten by 6-round decryption. So rk_{23}', rk_{24}, rk_{25}', rk_{26}, rk_{27}', and rk_{28} should be guessed and denoted by K_{guess}. The attack process is described as follows:

(1) Collect N pairs of plaintext (P, P') satisfying $P \oplus P' = (\Delta_0^4, \Delta_0^3, \Delta_0^2, \Delta_0^1)$ and their corresponding ciphertext pairs (C, C').

(2) Initialize 2^{32} counters $\mathscr{V}[0], \mathscr{V}[1], \ldots, \mathscr{V}[2^{32} - 1]$ and reset them.

(3) For each plaintext pair (P, P'), check whether its ciphertext pair (C, C') satisfies $(C \oplus C')[63 : 32] = 0x00000000$. If the ciphertext pair meets the formula, $x = (C \oplus C')[31 : 0]$ and $\mathscr{V}[x] + +$.

(4) Initialize 2^{96} counters $\mathscr{V}_k[0], \mathscr{V}_k[1], ..., \mathscr{V}_k[2^{96} - 1]$ and reset them.

(5) Guess 96-bit key K_{guess}, and decrypt x to obtain X_{23}^2, $X_{23}^{2'}$, X_{23}^1, and $X_{23}^{1'}$. Check whether $X_{23}^2 \oplus X_{23}^{2'}$ and $X_{23}^1 \oplus X_{23}^{1'}$ are equal to Δ_{23}^2 and Δ_{23}^1. If the conditions are met, then let $y = X_{23}^2 \parallel X_{23}^{2'} \parallel X_{23}^1 \parallel X_{23}^{1'}$.

(6) Use y to calculate $X_{22}^4 \oplus X_{22}^{4'}$ and $X_{22}^3 \oplus X_{22}^{3'}$. If $X_{22}^4 \oplus X_{22}^{4'} = \Delta_{22}^4$ and $X_{22}^3 \oplus X_{22}^{3'} = \Delta_{22}^3$, $\mathscr{V}_k[K_{guess}] + +$.

(7) Set advantage a to be 51, which implies that the top 2^{45} absolute values in $\mathscr{V}_k$ are kept. For each remaining key, we guess the remaining 58-bit subkey and calculate the master key. Finally, we test the key by trail encryptions.

If set $N = 2^{60}$, then about 2^{28} pieces of data enter step (5), which means 2^{124} 6-round encryptions, that is, $2^{121.7}$ 29-round encryptions. The complexity of step (7) is 2^{103} 29-round encryptions. So the total time complexity of this attack is about $2^{121.73}$ 29-round encryptions.

The counters $\mathscr{V}_k$ require storing 2^{96} bytes, so the memory complexity for the attack is 2^{96} bytes. The data complexity is 2^{61}.

The success rate $P_S = \Phi((\sqrt{pNS_N} - \Phi^{-1}(1 - 2^{-a}))/\sqrt{S_N + 1}) = 98.71$ by [13].

4. Linear Attack on 25-Round NUX

Linear approximations of NUX are searched for in this section, and the 25-round key-recovery attack is performed on NUX using a 19-round linear approximation.

4.1. Linear Approximations of NUX. To search for linear approximations of NUX, how masks propagate through S-boxes should be taken into account. When a mask passes through an S-box, the linear approximation table (LAT) of the S-box is looked up to determine the output mask and bias. Algorithm 2 searches for linear approximations of NUX and notations used in this algorithm are depicted in Figure 4.

The bias of one round can be 1/2, which can be obtained from the structure of NUX; that is, all the 8 S-boxes are passive. So special $(n - 1)$-round linear approximations are chosen, which satisfy $(\Gamma_0^4 = \Gamma_0^1 = 0)$, and one-round linear approximation with bias 1 is added to catch n-round linear approximations. Similar to searching for differential characteristics, there is only one active S-box in the first

Table 7: 22-Round differential characteristic of NUX.

Rounds i	Δ_i^4	Δ_i^3	Δ_i^2	Δ_i^1	Probability
0	0400	0000	0000	0000	1
1	0000	0000	0080	0000	2^{-3}
2	0040	0001	0000	0000	2^{-2}
3	0000	0000	0010	8000	2^{-3}
4	0004	1000	0000	0000	2^{-2}
5	0000	0000	0002	0100	2^{-2}
6	0800	6002	0000	0000	2^{-5}
7	0000	0000	003b	0811	2^{-5}
8	c80c	2b30	0000	0000	2^{-7}
9	0000	0000	0480	4236	2^{-5}
10	2040	0600	0000	0000	2^{-3}
11	0000	0000	0005	2002	2^{-2}
12	8080	0003	0000	0000	2^{-2}
13	0000	0000	0040	8800	2^{-2}
14	0400	0030	0000	0000	2^{-2}
15	0000	0000	0008	4004	2^{-3}
16	0008	3000	0000	0000	2^{-2}
17	0000	0000	0000	0110	1
18	0000	4800	0000	0000	2^{-3}
19	0000	0000	0020	0201	2^{-2}
20	4000	0500	0000	0000	2^{-3}
21	0000	0000	0000	2020	1
22	0000	2080	0000	0000	*

round of $n-1$ rounds. Meanwhile, Γ_i^4 and Γ_i^3, which lie in the two branches on the left side, will not affect Γ_i^2 and Γ_i^1, the two branches on the right side, during propagation in NUX. So the linear propagation on the left or right is taken into consideration, and differences of the other two branches are set to 0. In this way, the number of active S-boxes is always no more than 4 in one round, which can make the absolute value of the linear approximation bias as large as possible. The results of the search algorithm are listed in Table 8.

Moreover, the minimal numbers of active S-boxes of 1~5-round linear approximations are shown in the design manuscript [12]. And the minimal number of active S-boxes is also considered, and the minimum active S-boxes of 3~5-round linear approximations are less than the ones given in [12], which are presented in Table 9.

4.2. Linear Attack on NUX. Utilizing obtained linear approximations, a key-recovery attack can be applied to 25-round NUX using a 19-round linear approximation with bias 2^{-30}, which is described in Table 10. The 19-round linear approximation is put from the 4th to the 22th round of NUX, extending 3 rounds both backward and forward. The 25-round key-recovery attack is shown in Figure 5.

According to the linear approximation, there are $\Gamma_3^4 = \Gamma_3^2 = \Gamma_3^1 = \Gamma_{22}^4 = \Gamma_{22}^3 = 0x0000$, $\Gamma_3^3 = 0x1040$, $\Gamma_{22}^2 = 0x044c$, and $\Gamma_{22}^1 = 0x24af$. Furthermore, the following formula can be gotten:

$$\begin{aligned}&\Gamma_3^4 \cdot X_3^4 \oplus \Gamma_3^3 \cdot X_3^3 \oplus \Gamma_2^2 \cdot X_3^2 \oplus \Gamma_3^1 \cdot X_3^1 \oplus \Gamma_{22}^4 \cdot X_{22}^4 \oplus \Gamma_{22}^3 \\ &\quad \cdot X_{22}^3 \oplus \Gamma_{22}^2 \cdot X_{22}^2 \oplus \Gamma_{22}^1 \cdot X_{22}^1 = \Gamma_3^3 \cdot X_3^3 \oplus \Gamma_{22}^2 \cdot X_{22}^2 \\ &\quad \oplus \Gamma_{22}^1 \cdot X_{22}^1 = \kappa\end{aligned} \tag{4}$$

where X_3^3 can be calculated through 3-round encryption by guessing 26-bit subkeys, which are denoted by K_{before} including rk_0', $rk_1[15, 13, 10, 8, 7, 5, 2, 0]$, and $rk_2'[15, 7]$. Besides, X_{22}^2 and X_{22}^1 are obtained by 3-round decryption, which involves 40-bit subkeys, namely, $rk_{22}'[15, 14, 13, 10, 7, 4, 2, 0]$, rk_{23}, and rk_{24}'. For the sake of simplicity, $rk_{22}'[15, 14, 13, 10, 7, 4, 2, 0]$ and rk_{23} are denoted by K_{mid}. In the following, the attack process is depicted.

(1) Collect N plaintext/ciphertext pairs.

(2) Initialize 2^{66} counters $\mathscr{V}_0[0], \mathscr{V}_0[1], \ldots, \mathscr{V}_0[2^{66}-1]$ and reset them.

(3) Guess 26-bit key K_{before}.

(4) For each plaintext/ciphertext pair, calculate $x = X_3^3[12, 6] \parallel C^4 \parallel C^3 \parallel C^2 \parallel C^1$. Then $\mathscr{V}_0[x] + +$.

(5) Initialize 2^{34} counters $\mathscr{V}_1[0], \mathscr{V}_1[1], \ldots, \mathscr{V}_1[2^{34}-1]$ and reset them.

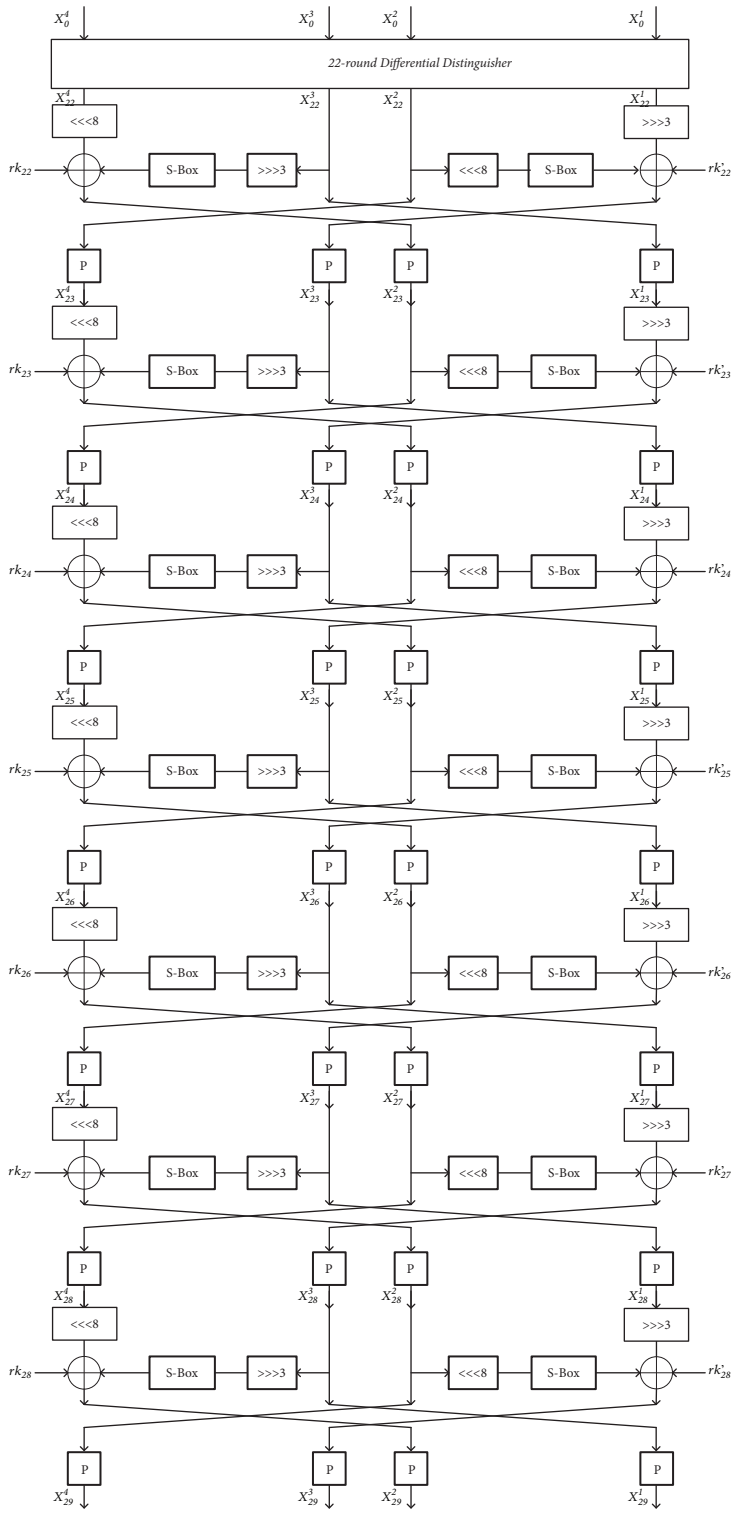


Figure 3: Differential attack of 29-round NUX.

(6) Guess 16-bit key rk'_{24}.

(7) For every x, calculate $y = X_3^3[12,6] \parallel X_{24}^2 \parallel X_{24}^1$. Then $\mathcal{V}_1[y] += \mathcal{V}_0[x]$.

(8) Initialize 2^{66} counters $\mathcal{V}_k[0], \mathcal{V}_k[1], \ldots, \mathcal{V}_k[2^{66}-1]$ and reset them.

(9) Guess 24-bit key K_{mid}.

(10) For every y, calculate $X_{22}^2[10,6,3,2]$ and $X_{22}^2[13,10,7,5,3,2,1,0]$. Then compute $z = \Gamma_3^3 \cdot X_3^3 \oplus \Gamma_{22}^2 \cdot X_{22}^2 \oplus \Gamma_{22}^1 \cdot X_{22}^1$. If $z = 0$, then $\mathcal{V}_k[K_{before} \parallel rk'_{24} \parallel K_{mid}] += \mathcal{V}_1[y]$; otherwise, decrease the counter by $\mathcal{V}_1[y]$.

Input: S-box S, Bias threshold $\epsilon Lim[30]$.
Output: A linear approximation with the best bias.
Data: The number of active S-boxes in i rounds $S_{active}[i]$.
The number of active S-boxes of the i-th round S^i_{active}.
(1) Generate the LAT of S-box.
(2) Store all 2, 4 and the corresponding input/output masks in LAT in the table LAT_P
(3) **for** Each of 8 S-boxes in the first round **do**
(4) **for** all non-zero entities in LAT **do**
(5) **if** S-box is in F_2 **then**
(6) $\Gamma^4_1 = \Gamma Y^h_1 \ggg 8, \Gamma^3_1 = \Gamma^2_1 = \Gamma^1_1 = 0$.
(7) Store them in $R_{in}[1]$.
(8) **else**
(9) $\Gamma^1_1 = \Gamma Y^l_1 \lll 3, \Gamma^4_1 = \Gamma^3_1 = \Gamma^2_1 = 0$.
(10) Store them in $R_{in}[1]$.
(11) Calculate $\Gamma^4_0, \Gamma^3_0, \Gamma^2_0, \Gamma^1_0$, and store them in $R_{in}[0]$.
(12) Calculate $\Gamma^4_2, \Gamma^3_2, \Gamma^2_2, \Gamma^1_2$, and store them in $R_{in}[2]$.
(13) Update $S_{active}[1]$ to be 1.
(14) The bias is recorded as $\epsilon[1]$.
(15) **for** $i \longleftarrow 2$ to 30 **do**
(16) **if** $\Gamma^4_i = \Gamma^3_i = 0$ **then**
(17) $\Gamma Y^l_i = \Gamma^1_i \ggg 3$
(18) Travel LAT_P to get ΓX^l_i, S^i_{active} and its corresponding bias ϵRnd
(19) **else if** $\Gamma^2_i = \Gamma^1_i = 0$ **then**
(20) $\Gamma Y^h_i = \Gamma^4_i \lll 8$
(21) Travel LAT_P to get ΓX^h_i, S^i_{active} and the corresponding bias ϵRnd
(22) Compute the total bias, and keep ones greater than threshold $\epsilon Lim[i]$
(23) Calculate $\Gamma^4_{i+1}, \Gamma^3_{i+1}, \Gamma^2_{i+1}, \Gamma^1_{i+1}$, and store them in $R_{in}[i+1]$.
(24) Update the number of active S-boxes $S_{active}[i]$ and bias $\epsilon[i]$
(25) **for** Each of 8 S-boxes in the first round **do**
(26) **for** all non-zero entities in DDT **do**
(27) Find the largest $\epsilon[i]$ and store it in $\epsilon Max[i]$
(28) Output linear approximations with bias $\epsilon Max[i]$

Algorithm 2: Algorithm for linear approximations on NUX.

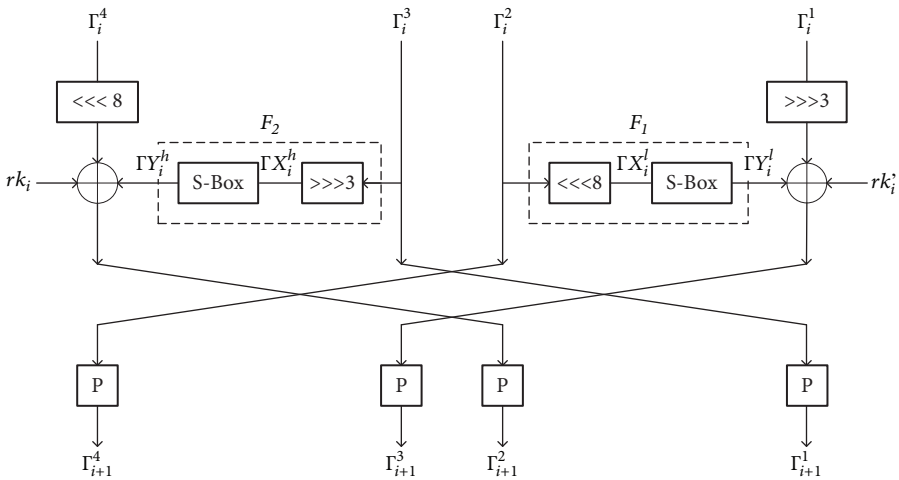


Figure 4: Linear representation of NUX.

(11) Set the advantage a to be 17, which means that the top 2^{49} absolute values in $\mathcal{V}_k$ are kept. For each remaining key, we guess 77-bit subkey to determine the master key. And then we test the key by trail encryptions.

If $N = 2^{63.7}$, then the time complexity of step (3), step (5), and step (7) is about $2^{89.7}$ 3-round encryption, 2^{108} 1-round decryption, and 2^{100} 2-round encryption, respectively. Besides, the complexity of step (11) is 2^{126} 25-round

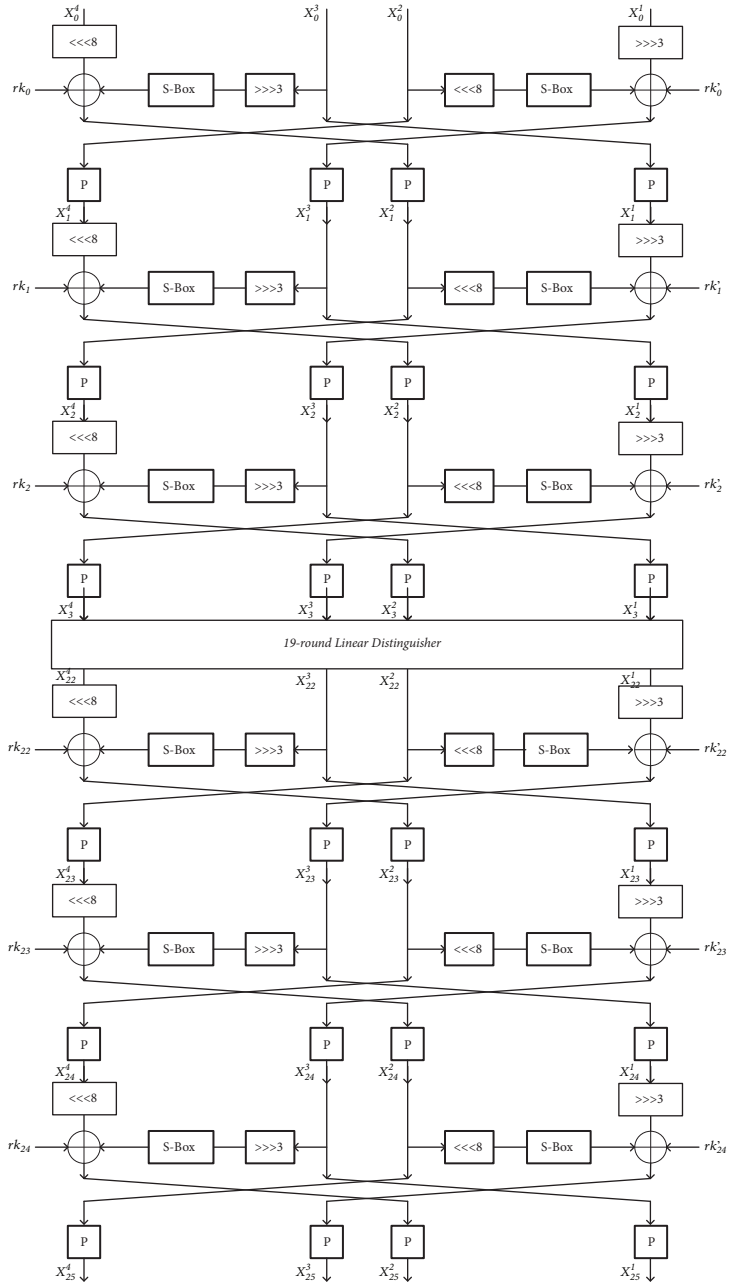


Figure 5: Linear attack of 25-round NUX.

encryption. Hence, the total time complexity of this attack is about 2^{126} 25-round encryption.

Both the counters $\mathscr{V}_0$ and $\mathscr{V}_k$ need 2^{69} bytes to store, so the memory complexity for the attack is 2^{70} bytes. The data complexity is $2^{63.7}$.

The success rate $P_S = \Phi(2\sqrt{N} \cdot |\epsilon| - \sqrt{1 + N/2^n}\Phi^{-1}(1 - 2^{-a-1})) = 88.21$ by [14].

5. Distinguishing Attack on NUX

Generally speaking, the distinguishing attack is a kind of test algorithm, which tries to perform the nonrandom behavior in cryptographic system. A distinguishing attack needs to find a distinguisher, which makes cryptographic algorithm different from random permutation. When analyzing NUX, we find a distinguisher with probability 1, that is, a deterministic distinguisher to distinguish NUX from a random permutation.

In Section 3, it has been pointed out that the two branches on the right side will not affect the two on the right side during difference propagation in NUX. Then, for the full-round NUX, when the input difference $(\Delta_0^4, \Delta_0^3, \Delta_0^2, \Delta_0^1)$ is $(0, 0, *, *)$, the output difference $(\Delta_{31}^4, \Delta_{31}^3, \Delta_{31}^2, \Delta_{31}^1)$ satisfies the form of $(*, *, 0, 0)$, shown in Figure 6, that is,

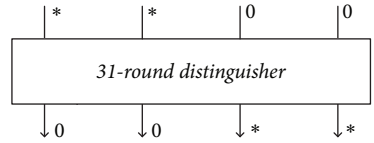


Figure 6: 31-Round differential distinguisher of NUX.

Table 8: Linear approximations of NUX.

Rounds	Bias	Number of tails
2	2^{-2}	288
3	2^{-3}	300
4	2^{-4}	13
5	2^{-5}	2
6	2^{-6}	4
7	2^{-10}	13
8	2^{-11}	62
9	2^{-12}	6
10	2^{-15}	70
11	2^{-16}	11
12	2^{-18}	70
13	2^{-19}	4
14	2^{-20}	2
15	2^{-22}	4
16	2^{-24}	16
17	2^{-26}	4
18	2^{-27}	4
19	2^{-30}	16
20	2^{-31}	4
21	2^{-33}	8
22	2^{-35}	60
23	2^{-37}	21
24	2^{-39}	25
25	2^{-41}	48
26	2^{-43}	103
27	2^{-44}	4
28	2^{-46}	7
29	2^{-48}	54
30	2^{-49}	4
31	2^{-50}	2

Table 9: Minimal number of active S-boxes from linear approximation.

Reference	Number of active S-boxes				
	Rounds				
	1	2	3	4	5
[12]	0	1	4	9	13
Section 4.1	0	1	2	3	3

$(0,0,*,*) \xrightarrow[31\text{-}round]{Pr=1} (*,*,0,0)$. However, the probability of the output difference to be $(*,*,0,0)$ is $Pr_{random} = 2^{-32}$ for random permutations, when the input difference is $(0,0,*,*)$. So 4 pairs of plaintexts are chosen, which are (P_i, P_i'), $(0 \le i \le 3)$, and $P_i \oplus P_i' = (0,0,*,*)$, and the corresponding ciphertexts (C_i, C_i') are checked to determine whether they satisfy $C_i \oplus C_i' = (*,*,0,0)$. The probability of obtaining such input/output differences is 1 for NUX, while it is 2^{-128} for a random permutation. Therefore, we can distinguish NUX from a random permutation. Besides, there is another distinguisher with probability 1, which is $(*,*,0,0) \xrightarrow[31\text{-}round]{Pr=1} (0,0,*,*)$ and can be used to perform a distinguishing attack like the one described before. So we will not explore it here.

Since only 4 pairs of plaintexts are used in the distinguishing attack, the data complexity is 8. And the attack needs no storage. In other words, the complexity of memory is 0. The time complexity is 8 31-round encryptions.

6. Conclusions

NUX is a 31-round iterative ultralightweight cipher, which is suitable for extremely constrained environment and is applied to the Internet of Things. In this paper, differential and linear trails are searched for 1~31-round NUX, which are better than those proposed in design specification. Moreover, a key-recovery attack on 29-round NUX is given with the 22-round differential characteristic found in the paper, whose time, data, and memory complexities are $2^{121.73}$ 29-round encryptions and 2^{61} and 2^{96} bytes, respectively. Meanwhile, using 22-round differential characteristic obtained in the paper, 29-round differential attack is performed with time, data, and memory complexities to be 2^{126} 25-round encryptions and $2^{63.7}$ and 2^{70} bytes, respectively. Furthermore, a distinguishing attack can be implemented on full NUX with data complexity 8. Results in this paper are the best ones on NUX till now.

Data Availability

All the data are obtained by our programs and can be provided to interested readers by email.

Conflicts of Interest

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

Acknowledgments

This work has been supported by National Cryptography Development Fund (no. MMJJ20170102), the National

Table 10: 19-Round linear approximation of NUX.

Rounds i	Γ_i^4	Γ_i^3	Γ_i^2	Γ_i^1	Bias
0	0000	1040	0000	0000	2^{-1}
1	0000	0000	0000	0500	2^{-2}
2	0001	4040	0000	0000	2^{-2}
3	0000	0000	0020	0600	2^{-2}
4	5100	0440	0000	0000	2^{-4}
5	0000	0000	8404	2028	2^{-3}
6	100a	a080	0000	0000	2^{-3}
7	0000	0000	0206	3000	2^{-3}
8	000a	2002	0000	0000	2^{-2}
9	0000	0000	0202	0800	2^{-2}
10	000a	0020	0000	0000	2^{-2}
11	0000	0000	0202	4010	2^{-3}
12	a000	0a00	0000	0000	2^{-2}
13	0000	0000	4040	0200	2^{-2}
14	1400	0400	0000	0000	2^{-3}
15	0000	0000	0084	0448	2^{-3}
16	01c3	8048	0000	0000	2^{-4}
17	0000	0000	9023	0180	2^{-2}
18	d800	4004	0000	0000	2^{-3}
19	0000	0000	044c	24af	*

Natural Science Foundation of China (nos. 61572293, 61502276, and 61692276), the National Natural Science Foundation of Shandong Province, China (ZR2016FM22), Major Scientific and Technological Innovation Projects of Shandong Province, China (2017CXGC0704), and Fundamental Research Fund of Shandong Academy of Sciences (no. 2018:12-16).

References

[1] A. A. Bogdanov, L. R. Knudsen, G. Leander et al., "An ultra-lightweight block cipher," in *Proceedings of the Cryptographic Hardware and Embedded Systems (CHES 2007)*, P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *Lecture Notes in Computer Science LNCS*, pp. 450–466, Springer, 2007.

[2] W. Wu and L. Zhang, "LBlock: a lightweight block cipher," in *Proceedings of the 9th International Conference on Applied Cryptography and Network Security (ACNS 2011)*, J. Lopez and G. Tsudik, Eds., vol. 6715 of *Lecture Notes in Computer Science*, pp. 327–344, Springer, Heidelberg, Germany, 2011.

[3] J. Borghoff, A. Canteaut, T. Güneysu et al., "PRINCE-A low latency block cipher for pervasive computing applications," in *Proceedings of the the 25th Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2012)*, X. Wang and K. Sako, Eds., vol. 7658 of *Lecture Notes in Computer Science*, pp. 208–225, Springer, Heidelberg, Germany, 2012.

[4] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[5] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT 1993)*, T. Helleseth, Ed., vol. 765 of *Lecture Notes in Computer Science*, pp. 386–397, Springer, 1993.

[6] A. G. Bafghi, R. Safabakhsh, and B. Sadeghiyan, "Finding the differential characteristics of block ciphers with neural networks," *Information Sciences*, vol. 178, no. 15, pp. 3117–3131, 2008.

[7] H. M. Heys and S. E. Tavares, "Substitution-permutation networks resistant to differential and linear cryptanalysis," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 9, no. 1, pp. 1–19, 1996.

[8] G. Jakimoski and L. Kocarev, "Differential and linear probabilities of a block-encryption cipher," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 1, pp. 121–123, 2003.

[9] J. Kim and R. C.-W. Phan, "Advanced differential-style cryptanalysis of the NSA's Skipjack block Cipher," *Cryptologia*, vol. 33, no. 3, pp. 246–270, 2009.

[10] F. Sano, K. Ohkuma, H. Shimizu, and S. Kawamura, "On the security of nested SPN cipher against the differential and linear cryptanalysis," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 86, no. 1, pp. 37–46, 2003.

[11] B.-Z. Su, W.-L. Wu, and W.-T. Zhang, "Security of the SMS4 block cipher against differential cryptanalysis," *Journal of Computer Science and Technology*, vol. 26, no. 1, pp. 130–138, 2011.

[12] G. Bansod, S. Sutar, A. Patil, and J. Patil, "NUX: a lightweight block cipher for security at wireless sensor node level. World academy of science, engineering and technology," *International Journal of Bioengineering and Life Sciences*, vol. 5, no. 1, 2018.

[13] A. A. Selçuk, "On probability of success in linear and differential cryptanalysis," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 21, no. 1, pp. 131–147, 2008.

[14] A. Bogdanov and E. Tischhauser, "On the wrong key randomisation and key equivalence hypotheses in Matsui's algorithm 2," in *Proceedings of the International Workshop on Fast Software Encryption (FSE 2013)*, S. Moriai, Ed., vol. 8424 of *Lecture Notes in Computer Science*, pp. 19–38, Springer, 2014.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 5860286, 13 pages
https://doi.org/10.1155/2019/5860286

Research Article

CasCP: Efficient and Secure Certificateless Authentication Scheme for Wireless Body Area Networks with Conditional Privacy-Preserving

Yong Xie, Songsong Zhang, Xiang Li, Yanggui Li, and Yuan Chai

Department of Computer Technology and Application, Qinghai University, China

Correspondence should be addressed to Yanggui Li; liyanggui@126.com

Received 6 March 2019; Accepted 24 April 2019; Published 4 June 2019

Guest Editor: Mingwu Zhang

As the aging population of society continues to intensify, the series of problems brought about by aging is becoming more and more serious. Because the health problem of the elderly brings many social problems, people have paid close attention to it. Fortunately, as a typical smart healthcare system, wireless body area networks (WBANs) present quit nice medical care for people, especially the aged. However, personal health information is very sensitive. But, the common communication channel is used in WBANs and any malicious entity can initiate a security attack on WBANs. To ensure secure communication and privacy-preserving which are the premise of the sound development of WBANs, an improved and efficient certificateless authentication scheme with conditional privacy-preserving is proposed in this paper on the basis of analyzing the most recent presented certificateless authentication scheme for WBANs. The proposed scheme also provides batch authentication to decrease authentication and communication cost. A rigid security proof demonstrates that our proposed scheme resists every type of security attack and can provide condition privacy-preserving. The performance analysis shows that our proposed scheme has some advantages in computation and communication cost.

1. Introduction

Nowadays, the population growth rate in many countries around the world is decreasing. Most of these counties have gradually entered an aged society. World Health Organization (WHO) has predicted that human life expectancy will reach 75 year old in 2030, and about 80 million people will be 60 year old in America and 430 million in China by 2050 [1]. Sociologists have pointed out that the aging population structure will put tremendous pressure on all aspects in society, especially healthcare.

In order to provide comprehensive and accurate care for the elderly, researchers have launched various research on smart healthcare. With the rapid development of wearable sensors, especially health sensors, wireless body area networks (WBANs) have a profound significance for improving the health monitoring of the elderly [2]. Information technologies are used in WBANs and can be well applied to medical related services [3]. In WBAN, client's information, such as weight trend, diet attempt, food-intake, hematologic biochemical parameters, respiratory rate, cardiac status, blood data, etc., is transmitted to the corresponding medical service application providers (AP) by wireless communication from body sensors. The client's doctor will receive this information soon and provide timely treatment based on this information [4, 5]. The scenes of sensor nodes collect and send the client real-time physiological data to AP and the typical smart medical service based WBANs can be depicted as in Figure 1.

However, the security and privacy issues in WBANs are very serious and worthy of paid close attention. It is well known that private personal health information is very sensitive, which may cause serious problems such as family conflicts, corporate crisis, and even state instability [6]. The health data are sent to AP through insecure communication channel and suffered from intercepting, eavesdropping, modification, and other attacks with little problem. The security of health data is critical to the patient as a forged health data results in doctor's misdiagnosis and extremely may endanger

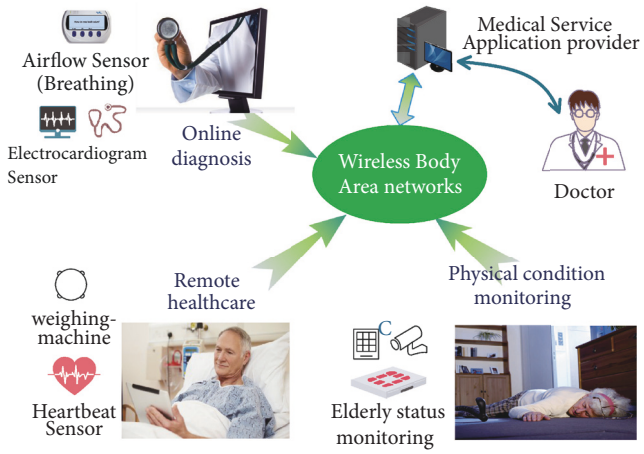


FIGURE 1: The typical smart medical services based WBANs.

the life of a patient. If WBANs cannot provide strong security protection measures, client's personal health information cannot be effectively protected, client will no longer trust WBANs, and people will no longer accept WBANs. Then, WBANs cannot get further development and cannot achieve the goal of smart medical care [7].

In order to meet the challenges of security and privacy protection in WBANs, many researchers have made continuous efforts and obtained some research results on WBANs authentication scheme. One important way is digit signature and data encryption. PKI-based authentication scheme and identity-based authentication scheme have been adopted to WBANs for a long time. But the PKI-based authentication scheme causes heavy certificate management and identity-based authentication scheme has an inevitable problem of key escrow. To solve this issue, certificateless authentication technology is introduced to WBANs and presents good application prospects. Recently, Ji *et al.* [8] proposed an efficient certificateless authentication scheme for WBANs. Ji *et al.* presented security analysis to show that their scheme can secure against all kinds of security attacks. However, their scheme cannot resist forgery attack and bath authentication attack, which is demonstrated in Section 5 in this paper. To the best of our knowledge, no universally accepted effective and secure authentication scheme for WBAN has been proposed, especially constructed by using certificateless public key cryptography [9]. Because of the strong privacy protection requirements of health data, limited communication channel, limited computing power, and fully open wireless communication environment, it is a huge challenge to build an efficient and secure certificateless authentication scheme for WBANs.

1.1. Motivations and Contributions. On reviewing Ji *et al.*'s certificateless authentication scheme [8], we decided to solve their security deficiencies while appreciating their high efficiency in message signing phase and authentication phase. In this paper, we present an improved and secure certificateless authentication scheme with conditional privacy-preserving (called CasCP). CasCP constructs signature and authentication algorithm by using elliptic curve cryptography (EEC) and no longer needs complex bilinear pairing operation. To sum up, there are three major contributions in our proposed scheme.

First, we present an improved and secure certificateless authentication scheme with conditional privacy-preserving. The proposed scheme includes five key phases for WBANs.

Second, we present a rigid security proof and detailed security analysis. It shows that CasCP can be secure against all known security attacks and providing privacy-preserving.

Third, the performance analysis shows that CasCP requires less computational and communication costs than recent similar schemes.

1.2. Organization of the Paper. The rest of the paper is arranged as follows. Related works and preliminaries are presented in Sections 2 and 3. Section 4 shows the system model and security requirements. Ji *et al.*'s certificateless authentication scheme is reviewed and analyzed in Section 5. Next, the CasCP is proposed in Section 6. Security proof and performance analysis are presented in Sections 7 and 8. At last, it draws a conclusion.

2. Related Works

In order to present a secure communication in WBANs, there are many security requirements. Among all of the security requirements, the remote authentication is the most basic and important requirement. In 1981, Lamport proposed the first remote authentication scheme [10] that allows the mobile user to authenticate with a server through a public channel and generate the session key to encrypt the later session. From then on, more and more remote authentication schemes have been proposed to apply to different environments.

Some works in [11–15] are constructed based on traditional public key cryptosystem (PKC). But there are many difficulties in the establishment, implementation, and management of traditional PKC system. In order to solve the problems in traditional PKC system for WBANs, some researchers have proposed mutual authentication scheme using identity-based public key cryptography [16, 17]. In this way, these authentication schemes solve the difficulties in traditional PKC system. However, there is another thorny problem, key escrow problem; that is, if the key generation center has been compromised, the system goes into a state of being out of control.

In 2003, Al Riyami *et al.* [18] proposed certificateless cryptography, which can erase key escrow problem in identity-based PKC. Based on the previous work [18], scholars have proposed a lot of secure authentication schemes [19, 20] by using certificateless cryptography. In 2005, Huang *et al.* [21] proposed an improved scheme over Al Riyami *et al.*'s [18] that can avoid security leaks. Huang *et al.* [20] proposed two certificateless signature schemes on assuming three-kind-adversary security model. However, it has been pointed out their scheme cannot resist key replacement attacks [22].

To decrease authentication and communication cost, Boneh *et al.* [23] presented a certificateless authentication

scheme with batch authentication in 2003. Batch authentication has been widely used for Internet of thing and other wireless networks, including WBANs. Without doubt, new security issues of batch authentication technology are unavoidable. Until now, researchers have proposed a lot of batch authentication schemes for WBANs and other wireless networks [24–26]. Based on the computational complexity of pairing, batch authentication and aggregate signature schemes [27–29] have been presented by using bilinear pairing. Xiong *et al.* [30] proposed an aggregate signature and batch authentication scheme that do not use clock synchronization and needs less computation cost than Zhang *et al.*'s [27] scheme. But, an adversary can successfully launch a forgery attack on Xiong et al.'s scheme [31, 32]. Wen *et al.* [33] constructed an aggregate signature scheme using bilinear pairing with designed verifier. Hartung *et al.* [34] presented another fault-tolerant batch authentication scheme. Tu *et al.* [29] proposed a revised authentication scheme to solve the security deficiencies of Xiong's scheme [30]. He *et al.* [35] presented a new certificateless authentication scheme for WBANs. Unfortunately, the foregoing schemes have more or less security deficiencies; some schemes cannot resist security attacks in batch authentication [36–38].

Most recently, Ji *et al.* [8] proposed a certificateless conditional privacy-preserving authentication scheme by using elliptic curve cryptography (ECC) for WBANs. Their proposed scheme has a clear advantage in computation performance when compared with the former certificateless scheme using bilinear pairing. They claimed that their proposed scheme provides conditional privacy-preserving and can resist all kinds of security attacks. However, we demonstrate that a common adversary can successfully launch a forgery attack in individual authentication and batch authentication. To solve the deficiencies of Ji *et al.*'s authentication scheme, we propose an improved certificateless authentication scheme with conditional privacy-preserving.

3. Preliminaries

3.1. Elliptic Curve Cryptosystem (ECC). In 1984, Miller proposed elliptic curve cryptography (ECC) for the first time [39]. Koblitz [40] proposed an ECC instance based on the difficulty of elliptic curve discrete logarithm problem (ECDLP) before long. Since then, researchers have proposed a lot of secure authentication schemes that are constructed with ECC since ECC is efficient to decrease computation cost [41]. The definition of ECC can be depicted as follows.

Let p be a large prime number; F_p is a finite field over p. Elliptic curve E/E_p meets equation $y^2 = x^3 + ax + b \bmod p$ with $a, b \in F_p$ and $(4a^3 + 27b^2) \bmod p \neq 0$. Let point Θ be an infinite point. Θ and other points in E/E_p form an addictive group G. Given P and Q are different points on E/E_p, $P+Q$ is defined as point addition. $m \cdot P = \underbrace{P + P + \cdots + P}_{m \text{ times}}$ is defined as scalar multiplication. n is defined as order if n is the smallest number that meets $n \cdot P = \Theta$.

3.2. Complexity Assumptions. Elliptic curve discrete logarithm problem (ECDLP): given two random points $P, Q \in E/E_p$ and $Q = x \cdot P$, without knowing $x \in_R Z_p^*$, it is hard to compute x from Q. The probability for an adversary A to solve the ECDLP problem is $Adv_A^{\text{ECDLP}} = \Pr[A(P, Q = x \cdot P) = x]$. The hardness is that to compute a from Q is negligible [42].

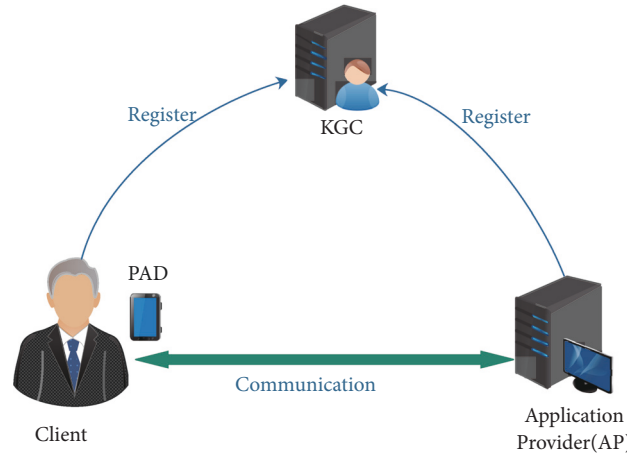


Figure 2: A common network structure of WBANs.

4. System Model and Security Model

4.1. System Model. There are three main entities in WBANs, i.e., key generation center (KGC), clients (including his/her PDA), and application providers (AP). KGC generates the system parameters and location public key and secret key for APs. Each client generates his/her secret value and then registers with KGC to obtain their public key, partial private key, and PDA with system parameter and partial key. At last, the clients could sign and send their messages to APs. In the process, AP and client should be authenticated each other and obtain an identical session key. The common network structure of wireless body area networks (WBANs) is illustrated in Figure 2.

Generally speaking, the messages in WBANs include sensitive health data. To ensure data integrity and identity authentication, these data should be signed and encrypted by PDA. The data with a signature could fall into two types: valid signature that can pass AP's authentication and invalid signature that cannot pass AP's authentication. When AP receives messages from different clients, AP can authenticate message one by one and also can adopt a more efficient way to authenticate multimessages, such as batch authentication. In our proposed scheme, batch authentication is used to improve authentication efficiency.

4.2. Security Model. In this section, we analyze the adversary model of certificateless authentication scheme for WBANs. As Al Riyami's work [18], two-level attacks exist in the certificateless PKC. One is type-I adversary (called $\mathcal{A}_{\mathcal{I}}$) who is able to simulate an "outsider" attacker; another is type-II adversary who is able to simulate an "insider" attacker (called $\mathcal{A}_{\mathcal{II}}$), who may be an "honest but curious" KGC. $\mathcal{A}_{\mathcal{I}}$ cannot get system secret key and users' partial key; however

Table 1: Notations used and description.

Symbol	Description
G	Acyclic group on an ECC with order q
P	The generator point of group G
s	The secret key of TA
P_{pub}	The public key of TA
RID_i	The real identity of a client
ID_A	The identity of a AP
PID_i	The pseudo identity of client
T	Validity period of pseudo identity
t	signature time
$H(.)$	One-way hash function

it could compromise users' secret value. $\mathscr{A}_{\mathscr{I}.\mathscr{I}}$ can get the system secret key and users' partial key but cannot get user secret value [43].

According to the ability of adversary $\mathscr{A}$ (include $\mathscr{A}_{\mathscr{I}}$ and $\mathscr{A}_{\mathscr{I}\mathscr{I}}$) and the system model of WBANs, we define security model as a game between a challenger $\mathscr{C}$ and $\mathscr{A}$ under the random Oracle model for the proposed scheme. Three steps are included in the game.

Initialization: $\mathscr{C}$ generates system parameters and system secret key. Then $\mathscr{C}$ gives the public parameters to $\mathscr{A}$.

Oracle query: $\mathscr{A}$ can make queries with h Oracle, *Create-User*, *Replace-Public-Key*, *Extract-Secret-Value*, *Extract-Partial-Key*, and *Sign* Oracle at will, unlimited query times and order. Then $\mathscr{C}$ answers $\mathscr{A}$ by the definition of game.

Output: $\mathscr{C}$ forges a signature after $\mathscr{A}$ has finished the above Oracle queries. At last, the advantage of successfully forging a valid signature is analyzed.

According to the definition of the game, $\mathscr{A}$ can breach the authentication scheme ϕ only if $\mathscr{A}$ could make a valid signature and pass authentication. Let $Adv_{\phi}^{Auth}(\mathscr{A})$ be the probability that $\mathscr{A}$ can breach ϕ during the game.

Definition 1. An authentication scheme for WBANs can be determined to be secure only if the probability $Adv_{\phi}^{Auth}(\mathscr{A})$ is negligible for any probabilistic-polynomial-time (PPT) adversary $\mathscr{A}$.

As definition of security requirement in most works for WBAN, we also agree that a secure certificateless authentication scheme for WBAN should provide anonymity, mutual authentication, traceability, and session key establishment; it also should be secure against modification attack, impersonation attack, replay attack, batch authentication attack, and other security attacks [44].

5. Review and Analysis of Ji *et al.*'s Scheme

In this section, we will review and analyze Ji *et al.*'s scheme [8]. To more clearly, Table 1 lists the notations and their descriptions adopted in Ji *et al.*'s scheme.

5.1. Review of Ji et al.'s Scheme. There are four phases in Ji *et al.*'s scheme [8], and the four phases can be briefly depicted as follows.

System Initialization Phase. TA executes this phase based on security parameter l.

(1) Choose two prime numbers p and q, define a finite field F_p, and then generate group G with order q on F_p.

(2) Let P be a generator of G, choose $s \in_R Z_q^*$, and compute $P_{pub} = sP$ as its public key. Then choose four one-way hash functions, $H_0, H_1, H_2, H_3 \longrightarrow Z_q^*$.

(3) Select $z_A \in_R Z_q^*$ for each registered AP and compute $b_A = z_A + sH_0(ID_A)$ as AP's private key and $B_A = z_A P$ as AP's public key.

Pseudo Identity Generation and Message Singing Phase. In this phase, each valid client should register with TA, then he/she can sign messages with his/her private key and send to AP. The detailed steps are as follows:

(1) The client chooses $r_i, x_i \in_R Z_q^*$, computes $X_i = x_i P$ and $PID_{i,1} = r_i P$, and then sends $\{RID_i, PW_i, PID_i, X_i\}$ to TA via a secure way.

(2) TA computes $\beta = H_0(RID_i) \oplus H_0(PW_i)$ and $PID_{i,2} = RID_i \oplus H_2(sPID_{i,1}, T_i)$, where T_i is the validity period of pseudo identity. Then TA chooses $w_i \in_R Z_q^*$ and computes $Y_i = w_i P$ and $y_i = w_i + s\alpha \bmod q$, where $\alpha_i = H_1(PID_i, X_i)$ and $PID_i = (PID_{i,1}, PID_{i,1}, T_i)$. Finally, TA loads $\{PID_i, y_i, \beta, Y_i\}$ into the client's PDA. Now, the client's private key is $SK_i = (x_i, y_i)$, and public key is $PK_i = (X_i, Y_i)$.

(3) Before signing a message, the client should input his RID_i and PW_i into his/her PDA. PDA checks whether it meets $\beta \stackrel{?}{=} H_0(RID_i) \oplus H_0(PW_i)$. If it does, the client can sign by using PDA as next step.

(4) Assume medical message be M_i; PDA chooses $d_i \in_R Z_q^*$ and timestamps t_i and computes $D_i = d_i P$, $u_i = H_3(M_i, PID_i, D_i, t_i)$, $\sigma_i = x_i + y_i + d_i \cdot u_i \bmod q$, and $K = d_i(B_A + H_0(ID_A)P_{pub})$, where K will be the session key between AP and the client. At last, PDA sends $\{PID_i, M_i, \sigma_i, D_i, t_i\}$ to AP.

Authentication Phase. AP can authenticate messages by the following two ways.

(i) Individual Authentication. When receiving a message $\{PID_i, M_i, \xi_i, D_i, t_i\}$, AP checks whether T_i and t_i are valid. If they do, AP computes $\alpha_i = H_1(PID_i, X_i)$ and $u_i = H_3(M_i, PID_i, D_i, t_i)$ and checks whether $\sigma_i P \stackrel{?}{=} X_i + Y_i + \alpha_i P_{pub} + u_i D_i$ holds or not. If it holds, AP accepts the message and computes session key $K = b_A D_i$ and then sends $MAC_K(B_A)$ to the client. At last, the client uses K as session key if the received $MAC_K(B_A)$ is identical to his/her $MAC_K(B_A)$.

(ii) Batch Authentication. When receiving n messages $\{PID_i, M_i, \xi_i, D_i, t_i\}_{i=1 \text{ to } n}$ from different clients, AP checks T_i and t_i for each message. Then AP computes $\alpha_i = H_1(PID_i, X_i)$ and $u_i = H_3(M_i, PID_i, D_i, t_i)$ for each message and checks whether the n messages meet the following equation:

$$\left(\sum_{i=1}^{n} \sigma_i\right) P = \sum_{i=1}^{n} X_i + \sum_{i=1}^{n} Y_i + \left(\sum_{i=1}^{n} \alpha_i\right) P_{pub} + \sum_{i=1}^{n} (u_i D_i) \tag{1}$$

If does, AP accepts these messages.

Password Change Phase. For security of PDA, the client can renew password PW_i locally by following steps.

(1) The client inputs RID_i and old password PW_i; the PDA checks $\beta \stackrel{?}{=} H_0(RID_i) \oplus H_0(PW_i)$. If it does, the PDA requires the client to input new password PW_i^*, then computes $\beta^* = \beta \oplus H_0(PW_i) \oplus H_0(PW_i^*)$, and replaces β with β^*.

5.2. Analysis of Ji et al.'s Scheme. In this subsection, the security deficiencies of Ji *et al.*'s scheme are analyzed.

(i) Not Be Secure against Forge Attack. Ji *et al.* show that their scheme could resist any forge attacks. However, any PPT $\mathscr{A}_{\mathscr{J}}$ could lightly win Game I in their scheme; that is, it could not be secure against forge attack. Assuming that the client's identity is PID_i, the adversary is $\mathscr{A}_{\mathscr{J}}$. $\mathscr{A}_{\mathscr{J}}$ launches the forge attack as the following steps:

(1) $\mathscr{A}_{\mathscr{J}}$ has intercepted or received a valid message $\{PID_i, M_i, \xi_i, D_i, t_i\}$, which meets verification function $\sigma_i P = X_i + Y_i + \alpha_i P_{pub} + u_i D_i$. Then $\mathscr{A}_I$ selects $w_i^* \in_R Z_q^*$, message M_i^*, and timestamps t_i^*.

(2) $\mathscr{A}_I$ computes $Y_i^* = w_i^* P - (X_i + \alpha_i P_{pub})$, $D_i^* = d_i^* P$, $u_i^* = H_3(M_i^*, PID_i, D_i^*, t_i^*)$, and $\sigma_i^* = w_i^* + u_i^* d_i^* \bmod q$, where PID_i is a valid pseudo identity. At last, $\mathscr{A}_I$ sends the forged message $\{PID_i, M_i^*, \xi_i^*, D_i^*, t_i^*\}$ to AP by using PID_i's identity.

(3) AP receives message $\{PID_i, M_i^*, \xi_i^*, D_i^*, t_i^*\}$, and checks whether the equation $\sigma_i^* P = X_i + Y_i^* + \alpha_i P_{pub} + u_i^* D_i^*$ holds or not. Let us expand the equation as follows:

$$\begin{aligned} X_i + Y_i^* + \alpha_i P_{pub} + u_i^* D_i^* &= X_i + Y_i^* + \alpha_i P_{pub} + u_i^* D_i^* \\ &= X_i + \left(-\left(\alpha_i P_{pub} + X_i\right) + w_i^* P\right) + \alpha_i P_{pub} + u_i^* D_i^* \\ &= Y_i^* + u_i^* D_i^* = \left(w_i^* + u_i^* d_i^*\right) P = \sigma_i^* P \end{aligned} \quad (2)$$

As shown above, $\mathscr{A}_{\mathscr{J}}$ can forge a valid message by using PID_i's identity easily. Therefore, Ji *et al.*'s scheme cannot resist any $\mathscr{A}_{\mathscr{J}}$'s forge attack.

(ii) Not Be Secure against Batch Authentication Attack. The adversary $\mathscr{A}_{\mathscr{J}}$ can also launch security attack in the batch authentication step of Ji *et al.*'s scheme. $\mathscr{A}_{\mathscr{J}}$ can do as the following steps.

(1) $\mathscr{A}_{\mathscr{J}}$ can forge two signatures $\sigma_1 = x_i + y_i$ and $\sigma_2 = u_i d_i$ on two messages $\{PID_i, M_i, \sigma_1, D_i, t_i\}$ and $\{PID_i, M_i, \sigma_2, D_i, t_i\}$, which cannot meet the verification. However, σ_1 and σ_2 can meet the batch authentication function of Ji *et al.*'s scheme as $\sum_1^2 \sigma_i P = \sum_1^2 X_i + \sum_1^2 Y_i + \sum_1^2 \alpha_i P_{pub} + \sum_1^2 u_i D_i = X_i + Y_i + \alpha_i P_{pub} + u_i D_i$.

Therefore, Ji *et al.*'s scheme cannot resist any $\mathscr{A}_{\mathscr{J}}$'s batch authentication attack.

6. The Improved Certificateless Authentication Scheme

In this section, an improved and secure certificateless authentication scheme for WBANs with conditional privacy-preserving (called CasCP) is proposed. The proposed CasCP consists of five phases: system initialization phase, pseudo identity generation phase, message signing phase, authentication phase, and password change phase.

Table 2: New notations and description in our scheme.

Symbol	Description
KGC	Key Generation center
$h(.)$	One-way hash function
$MAC_K(.)$	One-hash function with key K
l	Security-level parameter

To be clear, four new notations and descriptions that adopted in CasCP are listed in Table 2.

Next, the five phases are described as the following subsections.

6.1. System Initialization Phase. The KGC runs this phase with security level parameter l as follows.

(1) KGC chooses two prime numbers p and q and defines a finite field F_p and then generates group G with order q on F_p.

(2) Let P be one of generators of G. KGC chooses $s \in_R Z_q^*$ and computes $P_{pub} = sP$ as its public key. Then choose four one-way hash functions, $h_0, h_1, h_2, h_3 \longrightarrow Z_q^*$.

(3) KGC selects $z_A \in_R Z_q^*$ for each registered AP and computes $b_A = z_A + s \cdot h_0(ID_A)$ as AP's private key and $B_A = z_A P$ as AP's public key.

6.2. Pseudo Identity Generation Phase. Each WBANs client should register with KGC when he/her wants to obtain healthcare services. The client and KGC complete the pseudo identity phase as follow steps.

(1) Assume the client real identity be RID_i and his/her login password for PDA be PW_i. He/she chooses $x_i \in_R Z_q^*$, computes $X_i = x_i P$, and then sends $\{RID_i, PW_i, X_i\}$ to KGC via a secure way.

(2) Upon receiving $\{RID_i, PW_i, X_i\}$, KGC chooses $w_i \in_R Z_q^*$ and expiration time T_i and then computes $Y_i = w_i P$, $\beta = h_0(RID_i) \oplus h_0(PW_i)$, $PID_i = RID_i \oplus h_1(sX_i, T_i, Y_i)$, $\alpha_i = h_2(PID_i, X_i, Y_i, T_i)$, and $y_i = w_i + s \cdot \alpha \bmod q$. Finally, KGC loads $\{PID_i, y_i, \beta, Y_i, X_i, T_i\}$ into the client's PDA.

(3) When the client receives the PDA from KGC, he/she inputs RID_i and PW_i into PDA. Next PDA checks whether $\beta = h_0(RID_i) \oplus h_0(PW_i)$ holds or not. If it holds, the client's private key is $SK_i = (x_i, y_i)$ and public key is $PK_i = (X_i, Y_i)$. Otherwise, he/she registers again as next step.

6.3. Message Signing Phase. In this phase, the client signs messages by using PDA when he/she needs to communicate with others (such as AP) as the following steps.

(1) The client inputs RID_i and PW_i into PDA firstly. Then PDA checks whether $\beta = h_0(RID_i) \oplus h_0(PW_i)$ holds or not, where β is stored in the PDA. If it holds, the client can sign message by PDA.

(2) Assume the medical message be M_i. PDA chooses $d_i \in_R Z_q^*$ and timestamps t_i and then computes $D_i = d_i P$, $u_i = h_3(M_i, PID_i, X_i, T_i, Y_i, D_i, t_i)$, $\sigma_i = x_i + y_i + d_i \cdot$

Client and PDA		AP
Input RID_i, PW_i		
Check $\beta = ? h_0(RID_i) \oplus h_0(PW_i)$ $d_i \in Z_q^*$, $D_i = d_i P$ $u_i = h_3(M_i, PID_i, X_i, T_i, Y_i, D_i, T_i, t_i)$ $\sigma_i = x_i + y_i + d_i u_i \bmod q$ $K = d_i(B_A + h_0(ID_A)P_{pub})$ $msg_1 = \{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$	msg_1	Check T_i, t_i $\alpha_i = h_2(PID_i, X_i, Y_i, T_i)$ $u_i = h_3(M_i, PID_i, X_i, T_i, Y_i, D_i, T_i, t_i)$ $\sigma_i P = ? X_i + Y_i + \alpha_i P_{pub} + u_i D_i$ $K^* = b_A D_i$ $msg_2 = \{MAC_{K^*}(B_A)\}$
Check $MAC_K(B_A) = ? MAC_{K^*}(B_A)$	msg_2	

Figure 3: Message signing and individual authentication phase.

$u_i \bmod q$, and $K = d_i(B_A + h_0(ID_A)P_{pub})$, where K is the session key between AP and the client. At last, PDA sends $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ to AP.

6.4. Authentication Phase. To ensure the security of data, the client and AP can authenticate each other in the proposed scheme. In order to further improve the authentication efficiency, batch authentication is provided. Next, individual authentication and batch authentication are presented.

(i) Individual Authentication. (1) When receives a message $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ from the client, AP checks whether T_i and t_i valid. If they do, AP computes $\alpha_i = h_2(PID_i, X_i, Y_i, T_i)$ and $u_i = h_3(M_i, PID_i, X_i, T_i, Y_i, D_i, t_i)$ and checks whether the verification equation

$$\sigma_i P = X_i + Y_i + \alpha_i P_{pub} + u_i D_i \tag{3}$$

holds or not. If it holds, AP accepts the message and computes session key $K^* = b_A D_i$ and then sends $MAC_{K^*}(B_A)$ to the client.

(2) After receiving $MAC_{K^*}(B_A)$ from AP, the client uses his/her K that obtained in message signing phase to compute $MAC_K(B_A)$ and then checks whether the received $MAC_{K^*}(B_A)$ is identical to his/her $MAC_K(B_A)$. If it does, the client and AP have authenticated each other successfully and obtained an identical session key K for subsequent communications.

The message signing and individual authentication phase are illustrated as Figure 3.

(ii) Batch Authentication. When receiving n messages $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}_{i=1 \text{ to } n}$ from different clients, AP can execute batch authentication for the n messages.

(1) AP checks T_i and t_i for each message and then computes $\alpha_i = h_2(PID_i, X_i, Y_i, T_i)$ and $u_i = h_3(M_i, PID_i, X_i, T_i, Y_i, D_i, t_i)$ for each message.

(2) AP selects a small random integer vector $v = \{v_1, v_2, \ldots, v_n\}$, which have little computation cost in scalar multiplication [41].

(3) At last, AP checks whether the n messages meet the following equation:

$$\left(\sum_{i=1}^{n} v_i \sigma_i\right) P = \sum_{i=1}^{n} v_i X_i + \sum_{i=1}^{n} v_i Y_i + \left(\sum_{i=1}^{n} v_i \alpha_i\right) P_{pub} + \sum_{i=1}^{n} (v_i u_i D_i) \tag{4}$$

If it does, AP accepts these messages.

6.5. Password Change Phase. This phase is same as Ji *et al.*'s scheme, and the description will not be repeated here.

7. Security Proof and Analysis

In this section, a formal security proof of CasCP is presented. It shows that CasCP is unforgeable against adversary $\mathcal{A}$ (included $\mathcal{A}_{\mathcal{I}}$ and $\mathcal{A}_{\mathcal{II}}$), and CasCP can meet the security requirements of WBANs.

7.1. Security Proof. Next, CasCP is assessed on the security under the random Oracle model.

Theorem 2. *Assume $\mathcal{A}_{\mathcal{I}}$ be a PPT adversary who could win Game I with nonnegligible probability. Let $\mathcal{C}$ be a challenger who could solve ECDLP problem on advantage $\varepsilon_1 \geq (1 - q_{h_1}/q)^{q_c}(1 - 1/q_c)^{q_e}(1 - q_{h2}/q)(1 - q_{h3}/q)(1/q_c)\varepsilon$, where $q_{h_1}, q_{h_2}, q_{h_3}, q_c, q_e$ are the times of executing h_1, h_2, h_3, Create-User, and Extract-Partial-Key Oracle query, respectively.*

Proof. Let $\mathcal{A}_{\mathcal{I}}$ be a PPT adversary, which attempts to forge target client ID_o's valid message. $\mathcal{A}_{\mathcal{I}}$ could win Game-I with

a probability ε. Given an ECDLP instance $(G, P, Q = sP)$, $\mathscr{C}$ runs $\mathscr{A}_{\mathscr{J}}$ as a subroutine to solve the ECDLP instance. □

Step 1. $\mathscr{C}$ executes system initialization, and publics the parameters to A_I, given an ECDLP instance $(G, P, Q = P_{pub})$ to $\mathscr{C}$, from which it tries to compute s from P_{pub}.

Step 2. $\mathscr{A}_{\mathscr{J}}$ executes Oracle queries within limited query times, then $\mathscr{C}$ will answer $\mathscr{A}_{\mathscr{J}}$ as the following rules.

(i) Hash-Queries. $\mathscr{C}$ answers $\mathscr{A}_{\mathscr{J}}$ when he/she executes Oracle queries as follows.

(a) h_1*-Query*. As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID, X), $\mathscr{C}$ looks for (ID, X) in list L_{h_1}. If L_{h_1} has the entry, $\mathscr{C}$ returns τ_{h_1} to $\mathscr{A}_{\mathscr{J}}$. Otherwise, $\mathscr{C}$ chooses τ_{h_1}, x, T at random and sets $\tau_{h_1} \longleftarrow h_1(u \cdot Q, X, T)$. Finally $\mathscr{C}$ returns τ_{h_1} to $\mathscr{A}_{\mathscr{J}}$.

(b) h_2*-Query*. As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID), $\mathscr{C}$ looks for (ID) in list L_α. If L_α has the entry, $\mathscr{C}$ returns τ_α to $\mathscr{A}_{\mathscr{J}}$. Otherwise, $\mathscr{C}$ chooses $w \in Z_q^*$ at random and computes $Y = wP$ and sets $\tau_\alpha \longleftarrow h_2(ID, Y, X, T)$. Then $\mathscr{C}$ returns τ_α to $\mathscr{A}_{\mathscr{J}}$.

(c) h_3*-Query*. As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID, M), $\mathscr{C}$ looks for (ID, M) in sign list L_s. If L_s has the entry, $\mathscr{C}$ returns τ_u to $\mathscr{A}_{\mathscr{J}}$. Otherwise, $\mathscr{C}$ chooses $d \in Z_q^*$ at random, computes $D = dP$, and sets $\tau_u \longleftarrow h_3(m, PID, X, T, Y, D, t)$, where PID, X, T, and Y can be obtained from other h queries and create-user query. Then $\mathscr{C}$ returns τ_u to $\mathscr{A}_{\mathscr{J}}$.

(ii) Create-User(ID). As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID), $\mathscr{C}$ looks for (ID) in user list L_u. If L_u has an entry with ID, $\mathscr{C}$ returns X_{ID} to $\mathscr{A}_{\mathscr{J}}$. Otherwise, $\mathscr{C}$ randomly selects $x_{ID}, w_{ID}, h1_{ID} \in Z_q^*$ and computes $PID = ID \oplus h1_{ID}$, $PK_{ID} = w_{ID} \cdot P + \alpha \cdot Q$, and $y_{ID} = \perp$. Next, $\mathscr{C}$ adds $\{ID, x_{ID}, w_{ID}, Y_{ID}, X_{ID}, PK_{ID}\}$ to the corresponding list L_u, L_{h1}, L_{h2}.

(iii) Replace-Public-Key(ID). As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID), $\mathscr{C}$ chooses $x \in Z_q^*$ at random and computes $X = x \cdot P$, Finally, $\mathscr{C}$ adds (x, X) in L_u and sends (x, X) to $\mathscr{A}_{\mathscr{J}}$. As for y, $\mathscr{C}$ returns $\perp$.

(iv) Extract-Secret-Value(ID). As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID), $\mathscr{C}$ looks for user list L_u. If L_u has the entry, $\mathscr{C}$ returns x to $\mathscr{A}_{\mathscr{J}}$.

(v) Extract-Partial-Key(ID). As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID), $\mathscr{C}$ looks for user list L_u. If $ID = ID_o$, $\mathscr{C}$ sends $\perp$ to $\mathscr{A}_{\mathscr{J}}$. Else if L_u has the entry with ID, $\mathscr{C}$ sends PK to $\mathscr{A}_{\mathscr{J}}$, else $\mathscr{C}$ runs *Create-User*(ID) query and sends PK to $\mathscr{A}_{\mathscr{J}}$.

(vi) Sign(ID, M). As $\mathscr{A}_{\mathscr{J}}$ executes this query with (ID, M), $\mathscr{C}$ looks for tuple (ID) in L_u. If $ID \neq ID_o$, $\mathscr{C}$ randomly selects $d \in Z_q^*$, computes $D = dP$, $u = h_3(M, PID, X, T, Y, D, t)$, and $\sigma = x + y + u \cdot d \bmod q$, and then adds d, D, u in L_s. If $ID = ID_o$, $\mathscr{C}$ selects $\alpha, d, u \in Z_q^*$ at randomly and computes $D = dP$, $X = \sigma \cdot P - Y - u\alpha \cdot Q - u \cdot D$ and then adds d, D to L_s. At last, $\mathscr{C}$ returns (σ, D) to $\mathscr{A}_{\mathscr{J}}$.

Step 3. Finally, $\mathscr{C}$ obtains a forged message (ID, M, D, σ) under certain restrictions that the $\mathscr{A}_{\mathscr{J}}$ never makes *Extract-Partial-key* query with ID and *Sign* query with (ID, M). If $ID \neq ID_o$, $\mathscr{C}$ stops the game. Otherwise, $\mathscr{C}$ looks for the corresponding entry in L_{h2}, L_u, L_s. If there is not the corresponding σ, it stops the game. Otherwise, σ meets the following equation:

$$\sigma \cdot P = X + Y + \alpha \cdot Q + ud \cdot P \tag{5}$$

$\mathscr{A}_{\mathscr{J}}$ can replay the game based on forgery lemma [45]; he/her could obtain another forged message (ID, M, D^*, σ^*) by selecting another σ^*, α^*, d^*.

$$\sigma^* \cdot P = X + Y + \alpha^* \cdot Q + u^* d^* \cdot P \tag{6}$$

According to (5) and (6), $\mathscr{C}$ could obtain the $s = (((\sigma - \sigma*) - (ud - u^*d^*))/(\alpha - \alpha^*)) \bmod q$; i.e., $\mathscr{C}$ could solve the ECDLP problem. Next, the probability of $\mathscr{C}$ which obtains the correct solution for $(P, Q = sP)$ is analyzed. If $\mathscr{C}$ has done successfully, two events must happen.

(i) $Ev1$: never stop the game.

(ii) $Ev2$: σ is valid.

Therefore, the advantage of $\mathscr{C}$ is $\varepsilon_1 = Pr[Ev1 \cap Ev2] = Pr[Ev1]\ Pr[Ev2 \mid Ev1]$. The occurrence probability of $Ev1$ could be gained in *Create-user*, *Extract-Partial-key*, and *Sign* Oracle query during the game. Therefore, it can obtain $Pr[Ev1] \geq (1 - q_{h_1}/q)^{q_c}(1 - 1/q_c)^{q_e}(1 - q_{h2}/q)(1 - q_{h3}/q)q)(1/q_c)$. Therefore, we can get $\varepsilon_1 \geq (1 - q_{h_1}/q)^{q_c}(1 - 1/q_c)^{q_e}(1 - q_{h2}/q)(1 - q_{h3}/q)(1/q_c)\varepsilon$.

Theorem 3. *Assume $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ is a PPT super type-II adversary who can succeed in Game-II with nonnegligible probability. Let $\mathscr{C}$ be a challenger who can solve the ECDLP problem with advantage $\varepsilon_2 \geq (1 - q_{h_1}/q)^{q_c}(1 - 1/q_c)^{q_r}(1 - 1/q_c)^{q_x}(1 - q_{h2}/q)(1 - q_{h3}/q)(1/q_c)\varepsilon$, where $q_{h_1}, q_{h_2}, q_{h_3}, q_c, q_r, q_x$ denote the times of executing h_1, h_2, h_3, Create-User, Replace-Public-Key, and Extract-Secret-Value Oracle query, respectively.*

Proof. Assume $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ is a type-II adversary, and $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ attempts to forge target client ID_o's valid message. Then $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ could win Game-II with probability ε. Given an ECDLP instance $(G, P, Q = x_o \cdot P)$, let $\mathscr{C}$ be a challenger. Next, $\mathscr{C}$ runs $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ as a subroutine to solve ECDLP problem. □

Step 1. $\mathscr{C}$ executes system initialization and public parameters and s to $\mathscr{A}_{\mathscr{J}\mathscr{J}}$. Assume $(G, P, Q = x_o \cdot P)$ is given an ECDLP instance; $\mathscr{C}$ tries to compute x_o from Q.

Step 2. $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ executes Oracle queries within limited query times, then $\mathscr{C}$ will answer $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ as the following rules.

(i) Hash-Queries. $\mathscr{C}$ answers $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ when he/she executes Oracle queries as follows.

(a) h_1*-Query*. As $\mathscr{A}_{\mathscr{J}\mathscr{J}}$ executes the query with (ID, X), $\mathscr{C}$ looks for (ID, X) in list L_{h_1}. If L_{h_1} has the entry, $\mathscr{C}$ returns τ_{h_1} to $\mathscr{A}_{\mathscr{J}\mathscr{J}}$. Otherwise, $\mathscr{C}$ chooses $\tau_{h_1}, w \in Z_q^*$ at random and computes $Y = wP$ and sets $\tau_{h_1} \longleftarrow h_1(s \cdot X, T, Y)$. Finally $\mathscr{C}$ returns τ_{h_1} to $\mathscr{A}_{\mathscr{J}\mathscr{J}}$.

(b) h_2-Query and h_3-Query. As $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ executes the two queries, $\mathcal{C}$ could answer $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ as he/she answers $\mathcal{A}_{\mathcal{I}}$ in Game I.

(ii) Create-User(ID). As $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ executes the query with (ID), $\mathcal{C}$ looks for user list L_u. If L_u has the entry, $\mathcal{C}$ returns TK_{ID}. Otherwise, if $ID = ID_o$, $\mathcal{C}$ chooses $w \in Z_*^q$ and $X \in G$ at random and current time T, calculates $Y = w \cdot P$, $PID = ID \oplus h_1(\alpha \cdot X, T, Y)$, $\alpha = h_2(PID, X, Y, T)$, and $y = w + s\alpha \bmod q$, and sets $x = \perp$. Then, $\mathcal{C}$ will add (ID, u, y, x, α) to list L_u, L_{h2}, respectively. If $ID \neq ID_o$, $\mathcal{C}$ chooses $u, x \in Z_*^q$ at random and current time T and calculates $U = u \cdot P$, $PID = ID \oplus h_1(\alpha u \cdot P, T)$, $\alpha = h_2(PID, X, Y, T)$, $y = w + \alpha s \bmod q$, and $X = x \cdot P$. Then, $\mathcal{C}$ will add $(ID, x, w, y, X, Y, \alpha)$ to corresponding list L_u, L_{h2}, respectively.

(iii) Replace-Public-Key(ID). As $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ executes the query with (ID), $\mathcal{C}$ will answer $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ as the following two cases: if $ID \neq ID_o$, $\mathcal{C}$ will answer $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ with the definition of Partial Key Generation and Private key Generation algorithm. If $ID = ID_o$, $\mathcal{C}$ chooses $w \in Z_q^*$ and $X \in G$ at random. Next $\mathcal{C}$ calculates $Y = w \cdot P$, $PID = ID \oplus h_1(sX, T, Y)$, $\alpha = h_2(PID, X, Y, T)$, and $y = w + s\alpha \bmod q$ and sets $x = \perp$. Finally, $\mathcal{C}$ sends (Y, X) to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$.

(iv) Extract-Secret-Value(ID). As $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ executes the query with (ID), $\mathcal{C}$ looks for it in user list L_u. If L_u has the entry with ID, $\mathcal{C}$ sends x to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$. Or, if $ID \neq ID_o$, $\mathcal{C}$ chooses $x \in_R Z_q^*$ at random and adds it to L_u as ID's secret value. Next $\mathcal{C}$ sends x to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$. If $ID = ID_o$, $\mathcal{C}$ returns $\perp$.

(v) Extract-Partial-Key(ID). As $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ executes the query with (ID), $\mathcal{C}$ looks for it in L_u. If L_u has the entry with ID, $\mathcal{C}$ sends y to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$. Otherwise, $\mathcal{C}$ will execute *Create-User*(ID) query and then sends y to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$.

(vi) Sign(ID, m). As $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ executes the query with (ID, M), $\mathcal{C}$ looks for it in L_u. If $ID \neq ID_o$, $\mathcal{C}$ chooses $d \in Z_q^*$ at random and current time t and then calculates $D = d \cdot P$, $u = h_3(M, PID, X, T, Y, D, t)$, and $\sigma = x + y + du \bmod q$. Next $\mathcal{C}$ adds d, D, σ in L_s and sends (D, σ) to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$. If $ID = ID_o$, $\mathcal{C}$ chooses $d \in Z_q^*$ at random and calculates $D = d \cdot Q$, $u = h_3(M, PID, X, T, Y, D, t)$, and $\sigma = x + y + du \bmod q$ and then adds σ, D to L_s. At last, $\mathcal{C}$ returns (σ) to $\mathcal{A}_{\mathcal{I}\mathcal{I}}$.

Step 3. At last, $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ obtains a forged message (ID, M, D, σ) under the constrain restrictions that $\mathcal{A}_{\mathcal{I}\mathcal{I}}$ never makes *Extract-Partial-Key* query with ID and *Sign* query with (ID, M). If $ID \neq ID_o$, $\mathcal{C}$ stops the game. Otherwise, $\mathcal{C}$ looks for the corresponding entry in L_{h2}, L_u, L_s. If there is not corresponding σ, $\mathcal{C}$ stops the game or σ meets the following authentication equation:

$$\sigma \cdot P = Q + Y + \alpha \cdot P_{pub} + ud \cdot Q \tag{7}$$

$\mathcal{A}_{\mathcal{I}\mathcal{I}}$ can replay the game based on forgery lemma [45]; he/she could obtain another forged messages (ID, M, D^*, σ^*) by selecting another u^*, d^*.

$$\sigma^* \cdot P = Q + Y + \alpha \cdot P_{pub} + u^* d^* \cdot Q \tag{8}$$

According to (7) and (8), $\mathcal{C}$ could obtain the $x_o = ((\sigma - \sigma^*)/(ud - u^* d^*)) \bmod q$; i.e., $\mathcal{C}$ could solve the ECDLP problem. Next, the probability that $\mathcal{C}$ gains the correct solution for the instance $(P, Q = X_o = x_o \cdot P)$ is analyzed. If $\mathcal{C}$ has been successful, two events must happen.

(i) $Ev1$: never abort the game.

(ii) $EV2$: σ is valid.

Therefore, $\mathcal{C}$'s advantage is $\varepsilon_1 = Pr[Ev1 \cap Ev2] = Pr[Ev1]\ Pr[Ev2 \mid Ev1]$. The probability of $Ev1$'s occurrence can be gained in *Create-user*, *Extract-Partial-key*, and *Sign* Oracle query during the game. Therefore, it can obtain $Pr[Ev1] \geq (1 - q_{h_1}/q)^{q_c}(1 - 1/q_c)^{q_r}(1 - 1/q_c)^{q_x}(1 - q_{h2}/q)(1 - q_{h3}/q)(1/q_c)$. Therefore, we can get $\varepsilon_2 \geq (1 - q_{h_1}/q)^{q_c}(1 - 1/q_c)^{q_r}(1 - 1/q_c)^{q_x}(1 - q_{h2}/q)(1 - q_{h3}/q)(1/q_c)\varepsilon$.

Now, we can draw a conclusion that CapCP can resist two-level adversary on the condition of the ECDLP assumption which is established.

7.2. Other Security Analyses. Next, we will analyze whether CapCP meets the security requirements of WBANs.

(i) Anonymity. In CasCP, a client's real identity is embedded his/her pseudo identity $PID_i = RID_i \oplus h_1(sX_i, T_i, Y_i)$. PID_i is generated by KGC, any adversaries cannot retrieve the real identity from PID_i because $sX_i = sx_iP$, $X_i = x_iP$, and $P_{pub} = sP$ make up a classic CDH problem. Therefore CapCP provides anonymity for clients.

(ii) Mutual Authentication. After receiving a message $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ from a client, AP checks the validity and integrity of the message according to individual authentication equation. If it holds, the message can be regarded as a valid message. The AP signs and returns reply message in the same way to the client, the AP can also be securely authenticated. Therefore, CasCP can satisfy mutual authentication for WBANs.

(iii) Traceability. The clients' real identities are embedded in PID by $PID_i = RID_i \oplus h_1(sX_i, T_i, Y_i)$. KGC is the only authenticated one that can retrieve the real identity from PID_i because only KGC knows the system secret key s. Therefore, CasCP provides identity traceability for KGC.

(iv) Modification Attack. Assume a forged message $\{M_i, PID_i, X_i, T_i, Y_i^*, D_i^*, t_i^*, \sigma_i^*\}$ is modified from a valid message $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ by an adversary, the verifier could easily distinguish the forged message $\{M_i, PID_i, X_i, T_i, Y_i^*, D_i^*, t_i^*, \sigma_i^*\}$ because the forged message cannot meet the authentication equation $\sigma_i P = X_i + Y_i + \alpha_i P_{pub} + u_i D_i$. Therefore, CasCP is secure against modification attack.

(v) Session Key Establishment. In CasCP, the client and AP have a session key as $K = d_i(B_A + H_0(ID_A)P_{pub})$. From the definition of K, $K = d_i b_A P$, $D_i = d_i P$, and $PK_A = b_A P$ are CDH problem instance. An adversary cannot compute a valid session key because of ECDLP assumption's hardness. Therefore, CasCP can achieve secure session key establishment.

Table 3: The execution time of cryptographic operations.

Operation	Abbreviations	Execute time
Scalar multiplication	T_m	2.576
Exponentiation operation	T_e	3.857
Bilinear pairing operation	T_p	4.163

(vi) Impersonation Attack. To impersonate a client, an adversary must generate a valid message $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ to meet the authentication equation $\sigma_i P = X_i + Y_i + \alpha_i P_{pub} + u_i D_i$. But the adversary cannot generate the valid a valid message according to Theorems 2 and 3. Therefore, CasCP is secure against impersonation attack.

(vii) Replay Attack. An adversary replays an old message $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ by new time t_i in $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i^*, \sigma_i\}$. However, AP can find that this message is invalid by verification equation $\sigma_i P = X_i + Y_i + \alpha_i P_{pub} + u_i D_i$ according to the ECDLP assumption's hardness. That is CasCP can be secure against replay attack.

(viii) Batch Authentication Attack. When an invalid message or more messages join in batch authentication process, CasCP uses a small random integer vector to break the inherent relationship among of the signatures of messages. Therefore, an adversary cannot use the invalid or forged messages to launch batch authentication attack.

(ix) Lost PDA Attack. To use PDA, the correct RID_i and PW_i must be input into PDA. However, the adversary cannot login the PDA without knowing RID_i and PW_i, even if the adversary has breached PDA and has got the data in PDA. However, the data is nothing useful for the adversary.

8. Performance Analysis

In this section, the performance analysis of computation and communication cost is presented among four authentication schemes for WBANs that are the proposed scheme (CapCP), Ji *et al.*'s scheme [8] (2018), He *et al.*'s scheme [7] (2017), and Wu *et al.*'s scheme [46] (2016).

It is important to be fair and objective for performance analysis. Therefore, we adopt the simulation results of cryptographic operation execution time in [8]. Their simulation environments are set as follows: operation system is Windows 8, hardware is formed with 2.50 CHz Intel Core i5-2450 CPU, memory is 8.00 GB, and PBC (pairing based cryptography) is used to run the related cryptographic operations. Table 3 lists the execution times of main time-consuming cryptographic operations; the other cryptographic operations, such as point addition operation and one-way hash function which are much less than scalar multiplication, are not included in the comparison.

8.1. Computation Cost Analysis. Next, the proposed CasCP is compared with three authentication schemes for WBANs in terms of computation cost in the client's message signing phase, AP's individual authentication phase, and AP's batch authentication phase.

In Wu *et al.*'s scheme, the computation cost of the client in message signing phase comprises three scalar multiplication and two exponentiation operations; the computation cost of the AP in individual authentication phase comprises three scalar multiplication, two exponentiation operations, and one bilinear pairing operation; the n messages computation cost of the AP in batch authentication phase comprises $3n$ scalar multiplication, $2n$ exponentiation operation, and n bilinear pairing operations.

In He *et al.*'s scheme, the computation cost of the client in message signing phase comprises four scalar multiplication operations; the computation cost of the AP in individual authentication phase comprises four scalar multiplications and one bilinear pairing operations; the n messages computation cost of the AP in batch authentication phase comprises $4n$ scalar multiplication and n bilinear pairing operations.

In Ji *et al.*'s scheme, the computation cost of the client in message signing phase comprises three scalar multiplication operations; the computation cost of the AP in the individual authentication phase comprises four scalar multiplications; the n messages computation cost of the AP in batch authentication phase comprises $n+3$ scalar multiplication operations.

The proposed CasCP scheme adds a random small integer vector in batch authentication to increase its security. But the increased computational overhead is small; therefore it will not be considered in the computation cost comparison. That is, CasCP's computation costs in different phase can be considered to be the same as Ji *et al.*'s scheme. Here it is not presented; please refer to the previous analysis for Ji *et al.*'s scheme.

On the results of Table 3, the total execution time of the three phases in the four schemes is drawn, shown in Table 4.

The computation cost times of the client in message signing phase of CasCP and Ji *et al.*' scheme are 7.728 *ms*, which decrease by 49% and 25% when compared with the corresponding computation time of Wu *et al.*'s scheme and He *et al.*' scheme. The computation cost time of AP in individual authentication phase is 7.728 *ms*, which decreases by 60% and 46% when compared with the corresponding computation time of Wu *et al.*'s scheme and He *et al.*' scheme. The more intuitive computation cost comparison of the two phases in the four schemes is shown in Figure 4.

The computation cost comparisons of AP in batch authentication phase (assume $n = 30$ messages) are illustrated in Figure 5. As shown in Figure 5, our proposed CasCP and Ji *et al.*' scheme take an advantage on computation cost than Wu *et al.*'s scheme and He *et al.*'s scheme.

According to the former computation cost analysis in batch authentication phase, the proposed CasCP and Ji *et al.*'s scheme have a clear advantage than the other two schemes. Figure 6 depicts the computation costs in batch authentication phase for the different number of messages of the four schemes. Therefore, CasCP and Ji *et al.*'s scheme are more efficient than Wu *et al.*'s scheme and He *et al.*'s scheme regardless of the number of messages.

Table 4: The computation cost comparison of the four schemes.

	message signing phase (client)	individual authentication phase (AP)
Wu's Scheme	$2T_e + 3T_m \approx 15.442ms$	$1T_p + 3T_m + 2T_e \approx 19.604$
He's Scheme	$4T_m \approx 10.304ms$	$1T_p + 4T_m \approx 14.446ms$
Ji's Scheme	$3T_m \approx 7.728ms$	$3T_m \approx 7.728ms$
CasCP	$3T_m \approx 7.728ms$	$3T_m \approx 7.728ms$

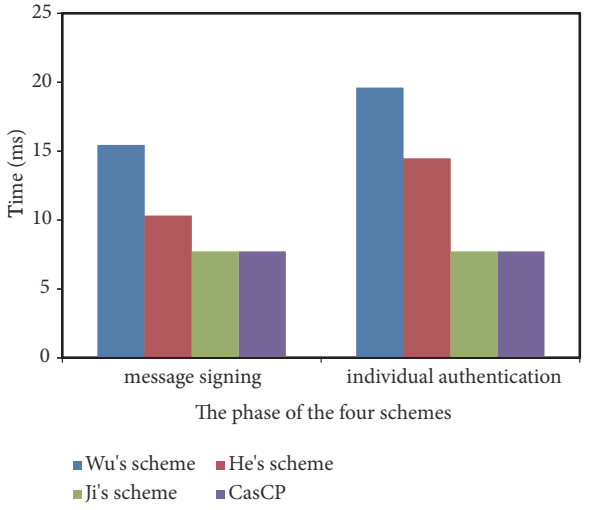


Figure 4: The computation costs of the two phases in the four schemes.

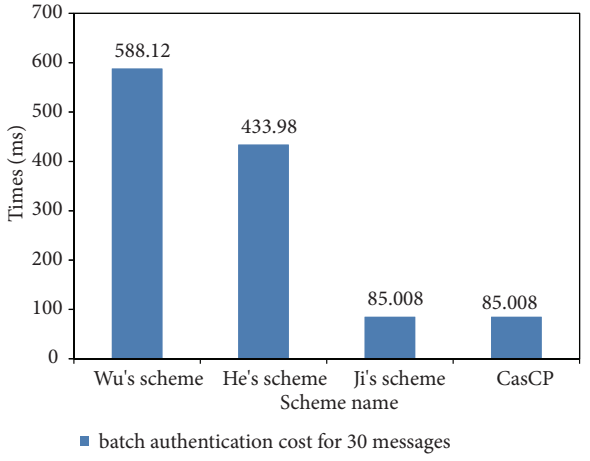


Figure 5: The computation costs of the four schemes for 30 messages in batch authentication phase.

In summary, compared with Wu *et al.*'s scheme and He *et al.*'s scheme, CasCP and Ji *et al.*'s scheme have lower computation cost in message signing phase, individual authentication phase, and batch authentication phase.

8.2. Communication Cost Comparison. In the subsection, we analyze the communication cost of the proposed CasCP and the three authentication schemes for WBANs in this subsection.

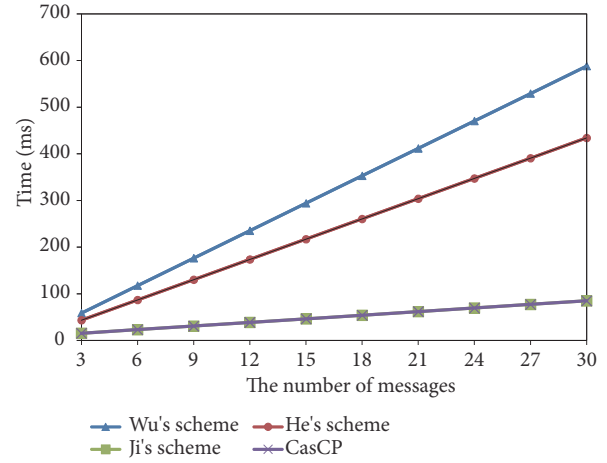


Figure 6: The computation costs of the four schemes for different number of messages in batch authentication.

According to the definitions of the above cryptographic operations, we assume that the size of p is 20 bytes, the element in G is 40 bytes, and the size of other communication elements in P is 20 bytes. For simplicity, message M_i is not included in the comparison.

In Wu *et al.*'s scheme, the message sent by a client to AP consists of $\{ID_c, V_i, auth_c, t_c\}$; the message sent by a client to AP consists of $\{R_{AP}, auth_{AP}, t_{AP}\}$. The messages include four elements in G, i.e., $(ID_c, V_i, R_{AP} \in G$, 40×3 bytes), and four elements in P, i.e., $(t_i, auth_c, T_{AP}, authe_{AP}$, 20×4 bytes); the total size of one communication round is 200 bytes.

In He *et al.*'s scheme, the message sent by a client to AP consists of $\{QID_i, T_i, t_i\}$; the message sent by a client to AP comprises $\{Y, auth_s\}$. The messages have four elements in G, i.e., $(QID_i, T_i, Y \in G$, 40×3 bytes), and two elements in P, i.e., $(t_i, auth_s$, 20×2 bytes); the total size of one communication round is 200 bytes.

In Ji *et al.*'s scheme, the message sent by a client to AP consists of $\{M_i, PID_i = \{PID_{i,1}, PID_{i,2}, X_i\}, T_i, Y_i, D_i, t_i, \sigma_i\}$, which has four elements in G, i.e., $(PID_{i,1}, X_i, Y_i, D_i \in G$, 40×4 bytes), and four elements in P, i.e., $(t_i, PID_{i,2}, T_i, \sigma_i$, 20×4 bytes). The message sent by a client to AP consists of $\{MAC_K(B_A)\}$. Therefore, the total size of one communication round is 240 bytes.

In CasCP, the message sent by a client to AP consists of $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$, which has three elements in G, i.e., $(X_i, Y_i, D_i \in G$, 40×3 bytes), and four elements in P, i.e., $(t_i, PID_i, T_i, \sigma_i$, 20×4 bytes). The message sent by a client to AP consists of $\{MAC_K(B_A)\}$; the size of $\{MAC_K(B_A)\}$ is defined

Table 5: The comparison of communication cost.

	Client-AP Component	Size
Wu *et al.*'s scheme	Client ⟶ AP $\{ID_c, V_i, auth_c, t_c\}$ AP ⟶ Client $\{R_{AP}, auth_{AP}, t_{AP}\}$	200 bytes
He *et al.*'s scheme	Client ⟶ AP $\{QID_i, T_i, t_i\}$ AP ⟶ Client $\{Y, auth_s\}$	180 bytes
Ji *et al.*'s scheme	Client ⟶ AP $\{M_i, PID_i = \{PID_{i,1}, PID_{i,2}, X_i\}, T_i, Y_i, D_i, t_i, \sigma_i\}$ AP ⟶ Client $\{MAC_K(B_A)\}$	240 bytes
CapCP	Client ⟶ AP $\{M_i, PID_i, X_i, T_i, Y_i, D_i, t_i, \sigma_i\}$ AP ⟶ Client $\{MAC_K(B_A)\}$	200 bytes

Tip: M_i is excluded in comparison.

as 20 bytes. Therefore, the total size of one communication round is 200 bytes.

The communication cost comparison results of one communication round between a client and AP are shown in Table 5. Compared with Wu *et al.*'s scheme and He *et al.*'s scheme, CasCP has no advantage because the two schemes decrease communication cost by encrypting data. Tip, the encrypted data must be more than 20 bytes that we assumed. Compared with Ji *et al.*'s scheme, CasCP scheme's communication cost has decreased by 15.4%.

Overall, CasCP scheme needs less computation cost than Wu *et al.*'s scheme and He *et al.*'s scheme. CasCP incurs less communication cost than Ji *et al.*'s scheme under the premise of solving the security deficiencies of Ji *et al.*'s scheme, providing conditional privacy-preserving and batch authentication.

9. Conclusion and Future Work

To provide privacy protection, Ji *et al.* proposed a certificateless authentication scheme for WBANs. However, the security analysis in this paper demonstrates that their scheme cannot be secure against forgery attack and batch authentication attack. To solve the deficiencies, an improved certificateless authentication scheme with conditional privacy-preserving (called CasCP) constructs signature with ECC, without any bilinear pairing operation. CasCP also provides batch authentication function and conditional privacy-preserving. A rigid security proof and analysis prove that CasCP is secure against different level adversary's attacks, such as adversary $\mathcal{A}_{\mathcal{I}}$ and adversary $\mathcal{A}_{\mathcal{II}}$. Compared with similar authentication scheme, CasCP has some advantages in computation and communication cost. Therefore, the proposed CasCP is more suitable for the WBANs.

Although CasCP is efficient and more secure than similar recent proposed schemes, more efficient authentication scheme is more favored, especially lightweight authentication scheme. Therefore, our next work is to study a secure lightweight authentication scheme with batch verification function for WBANs.

Data Availability

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

Conflicts of Interest

We declare that we do not have any commercial or associative interest that represents conflicts of interest in connection with the work submitted.

Acknowledgments

The work was supported in part by the National Natural Science Foundation of China under Grant 61862052 and the National Natural Science Function of Qinghai Province (2019-ZJ-7065 and 2017-ZJ-959Q).

References

[1] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1658–1686, 2014.

[2] N. Dey, A. S. Ashour, F. Shi, S. J. Fong, and R. S. Sherratt, "Developing residential wireless sensor networks for ECG healthcare monitoring," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 442–449, 2017.

[3] L. Wu, J. Wang, K.-K. R. Choo, and D. He, "Secure key agreement and key protection for mobile device user authentication," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 319–330, 2018.

[4] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: metrics, algorithms, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 550–586, 2017.

[5] P. Kumar, S. Kumari, V. Sharma, A. K. Sangaiah, J. Wei, and X. Li, "A certificateless aggregate signature scheme for healthcare wireless sensor network," *Sustainable Computing: Informatics and Systems*, vol. 18, no. 1, pp. 80–89, 2018.

[6] 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, 2018.

[7] D. He, S. Zeadally, N. Kumar, and J.-H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2590–2601, 2017.

[8] S. Ji, Z. Gui, T. Zhou, H. Yan, and J. Shen, "An efficient and certificateless conditional privacy-preserving authentication scheme for wireless body area networks big data services," *IEEE Access*, vol. 6, pp. 69603–69611, 2018.

[9] Q. Jiang, Z. Chen, B. Li, J. Shen, L. Yang, and J. Ma, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1061–1073, 2018.

[10] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[11] M. Li, S. Yu, J. D. Guttman, W. Lou, and K. Ren, "Secure ad hoc trust initialization and key management in wireless body area networks," *ACM Transactions on Sensor Networks*, vol. 9, no. 2, article no. 18, 2013.

[12] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223–230, 2012.

[13] A. S. Sangari and J. M. L. Manickam, "Public key cryptosystem based security in wireless body area network," in *Proceedings of the 2014 International Conference on Circuits, Power and Computing Technologies, ICCPCT 2014*, pp. 1609–1612, IEEE, Nagercoil, India, March 2014.

[14] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1615–1625, 2014.

[15] M. Li, W. J. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless Communications Magazine*, vol. 17, no. 1, pp. 51–58, 2010.

[16] X. Li, J. Niu, J. Liao, and W. Liang, "Cryptanalysis of a dynamic identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 374–382, 2015.

[17] C. C. Tan, S. Zhong, H. Wang, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 148–153, ACM, Alexandria, VA, USA, March-April 2008.

[18] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2894 of *Lecture Notes in Computer Science*, pp. 452–473, Springer, Taipei, Taiwan, 2003.

[19] G. Zheng, L. Yu, H. Xuan, and C. Kefei, "Two certificateless aggregate signatures from bilinear maps," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, pp. 188–193, IEEE, Qingdao, China, August 2007.

[20] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signatures: new schemes and security models," *The Computer Journal*, vol. 55, no. 4, pp. 457–474, 2012.

[21] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of a certificateless signature scheme," in *Proceedings of the International Conference on Cryptology and Network Security*, vol. 3810 of *Lecture Notes in Computer Science*, pp. 13–25, Springer, Xiamen, China, 2005.

[22] K.-A. Shim, "Security models for certificateless signature schemes revisited," *Information Sciences*, vol. 296, pp. 315–321, 2015.

[23] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 416–432, Springer, Warsaw, Poland, 2003.

[24] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," *Computer Communications*, vol. 32, no. 6, pp. 1079–1085, 2009.

[25] X. Liu, H. Zhu, J. Ma, Q. Li, and J. Xiong, "Efficient attribute based sequential aggregate signature for wireless sensor networks," *International Journal of Sensor Networks*, vol. 16, no. 3, pp. 172–184, 2014.

[26] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin, "Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 8, pp. 2562–2574, 2016.

[27] L. Zhang, B. Qin, Q. Wu, and F. Zhang, "Efficient many-to-one authentication with certificateless aggregate signatures," *Computer Networks*, vol. 54, no. 14, pp. 2482–2491, 2010.

[28] F. Zhang, L. Shen, and G. Wu, "Notes on the security of certificateless aggregate signature schemes," *Information Sciences*, vol. 287, pp. 32–37, 2014.

[29] H. Tu, D. He, and B. Huang, "Reattack of a certificateless aggregate signature scheme with constant pairing computations," *The Scientific World Journal*, vol. 2014, Article ID 343715, 10 pages, 2014.

[30] H. Xiong, Z. Guan, Z. Chen, and F. Li, "An efficient certificateless aggregate signature with constant pairing computations," *Information Sciences*, vol. 219, pp. 225–235, 2013.

[31] L. Cheng, Q. Wen, Z. Jin, H. Zhang, and L. Zhou, "Cryptanalysis and improvement of a certificateless aggregate signature scheme," *Information Sciences*, vol. 295, pp. 337–346, 2015.

[32] D. He, M. Tian, and J. Chen, "Insecurity of an efficient certificateless aggregate signature with constant pairing computations," *Information Sciences*, vol. 268, pp. 458–462, 2014.

[33] Y. Wen, J. Ma, and H. Huang, "An aggregate signature scheme with specified verifier," *Journal of Electronics*, vol. 20, no. 2, pp. 333–336, 2011.

[34] G. Hartung, B. Kaidel, A. Koch, J. Koch, and A. Rupp, "Fault-tolerant aggregate signatures," in *Public-Key Cryptography–PKC 2016*, vol. 9614, pp. 331–356, Springer, 2016.

[35] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, no. 99, pp. 1–10, 2015.

[36] L. Shen, J. Ma, X. Liu, and M. Miao, "A provably secure aggregate signature scheme for healthcare wireless sensor networks," *Journal of Medical Systems*, vol. 40, no. 11, article no. 244, 2016.

[37] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[38] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and K-anonymity," *World Wide Web*, pp. 1–11, 2018.

[39] J. G. Miller, "Culture and the development of everyday social explanation," *Journal of Personality and Social Psychology*, vol. 46, no. 5, pp. 961–978, 1984.

[40] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[41] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[42] Y. Xie, X. Li, S. Zhang, and Y. Li, "iclas: an improved certificateless aggregate signature scheme for healthcare wireless sensor networks," *IEEE Access*, vol. 7, pp. 15170–15182, 2019.

[43] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[44] Y. Liu, Y. Wang, X. Wang, Z. Xia, and J. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Computer Networks*, vol. 148, pp. 340–348, 2019.

[45] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 1070 of *Lecture Notes in Computer Science*, pp. 387–398, Springer, Saragossa, Spain, 1996.

[46] L. Wu, Y. Zhang, L. Li, and J. Shen, "Efficient and anonymous authentication scheme for wireless body area networks," *Journal of Medical Systems*, vol. 40, no. 6, article no. 134, 2016.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 8319508, 6 pages
https://doi.org/10.1155/2019/8319508

Research Article

Improved Cryptanalysis of a Fully Homomorphic Symmetric Encryption Scheme

Quanbo Qu,[1] **Baocang Wang**,[1,2] **Yuan Ping**,[2] **and Zhili Zhang**[2]

[1]*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China*
[2]*School of Information Engineering, Xuchang University, Xuchang 461000, China*

Correspondence should be addressed to Zhili Zhang; zlzhangxc@163.com

Received 5 March 2019; Accepted 22 April 2019; Published 2 June 2019

Guest Editor: Mingwu Zhang

Homomorphic encryption is widely used in the scenarios of big data and cloud computing for supporting calculations on ciphertexts without leaking plaintexts. Recently, Li et al. designed a symmetric homomorphic encryption scheme for outsourced databases. Wang et al. proposed a successful key-recovery attack on the homomorphic encryption scheme but required the adversary to know some plaintext/ciphertext pairs. In this paper, we propose a new ciphertext-only attack on the symmetric fully homomorphic encryption scheme. Our attack improves the previous Wang et al.'s attack by eliminating the assumption of known plaintext/ciphertext pairs. We show that the secret key of the user can be recovered by running lattice reduction algorithms twice. Experiments show that the attack successfully and efficiently recovers the secret key of the randomly generated instances with an overwhelming probability.

1. Introduction

With the rapid development of big data, the significance of privacy and security issues was highly regarded. A series of cryptoghraphic applications, such as fair electronic transaction [1], outsourcing data classification [2], lightweight security system of Internet of Things [3], mobile Ecommerce [4], and data mining based on homomorphic encryption, have been proposed.

Homomorphic encryption schemes allow users to meaningfully calculate ciphertexts without knowing the underlying plaintexts. For example, the RSA cryptosystem [5] (Pallier cryptosystem [6], respectively) only supports homomorphic multiplications (additions, respectively) on ciphertexts. In 2009, Gentry [7] designed the first fully homomorphic encryption scheme with ideal lattices. Thereafter, significant efforts had been performed to improve the efficiency of homomorphic encryption schemes [8–10]. However, all the known fully homomorphic encryptions are criticized for the high ciphertext expansion and ciphertext refreshing costs and hence cannot be directly used in practice. So researchers designed some cryptographic schemes with homomorphic properties dedicated to some concrete computing scenarios [11–15].

Recently, Li et al. [16] designed a symmetric homomorphic encryption scheme for outsourced databases that allow multiple data owners to efficiently share their data securely without compromising the privacy of the data. However, Wang et al. [17] observed that if some plaintext/ciphertext pairs were succesfully overdropped, one can efficiently recover the corresponding secret key of the scheme from the obtained plaintext/ciphertext pairs.

In practical scenarios, it may be difficult for the adversary to capture plaintext/ciphertext pairs. In this paper, we propose a new efficient cryptanalytic attack on Li et al.'s homomorphic encryption scheme. The attack consists of two stages. In the first stage, we separate the parts of the ciphertexts, which contain no secret key s. In the second stage, we separate the parts of the ciphertexts, which contain neither secret key s nor q. Thus s and q can be calculated during an acceptable time. The whole attack needs only several ciphertexts without corresponding plaintexts.

This paper is organised as follows. In Section 2, we review Li et al.'s symmetric homomorphic encryption scheme and

introduce the concept of lattice. In Section 3, we propose our attack and give the experimental results. In Section 4, we conclude our work.

2. Preliminaries

2.1. Notations. In this paper, the symbol $\mathbb{Z}$ is used to denote the ring of integer. Matrices are represented with bold upper-case characters like $\mathbf{B}$, while vectors are represented with bold lower-case characters like $\mathbf{v}$. All of the vectors in this paper are represented as row vectors. The symbol $\|\mathbf{v}\|$ means the length of vector $\mathbf{v}$ under the Euclidean norm, while the symbol $|p|_2$ means the bit length of integer p.

The symbol $\ll$ means "much less than", *i.e.* if $a \ll b$, the ratio a/b is a negligible function $negl(\lambda)$ of the security parameter λ. In mathematics, a negligible function $\mu(x)$ means that for any polynomial function $poly(x)$, there exists an integer N_c such that for any $x > N_c$,

$$|\mu(x)| < \frac{1}{poly(x)}. \tag{1}$$

2.2. Symmetric Homomorphic Encryption. The symmetric homomorphic encryption scheme proposed by Li et al. comprises these three algorithms as follows:

(i) *Key generation algorithm* $KeyGen()$

$$(s, q, p) \longleftarrow KeyGen(\lambda). \tag{2}$$

Input a security parameter λ, this algorithm outputs a secret key $SK = (s, q)$ and a public parameter p, where $q \ll p$.

(ii) *Encryption algorithm* $E()$

$$c = E(SK, m, d) = s^d (rq + m) \pmod p. \tag{3}$$

Input a secret key SK, a plaintext $m \in F_q$ and a parameter d, this algorithm outputs a ciphertext $c = E(SK, m, d)$. Notice that the parameter r should satisfy $|r|_2 + |q|_2 < |p|_2$.

(iii) *Decryption algorithm* $D()$

$$m = D(SK, c, d) = \left(c \times s^{-d} \pmod p\right) \pmod q. \tag{4}$$

Input a secret key SK, a ciphertext $c \in F_p$ and the ciphertext's degree d, this algorithm outputs a ciphertext $c \longleftarrow E(SK, m, d)$. The proof of the correctness is simple:

$$\begin{aligned} D(SK, c, d) &= \left(c \times s^{-d} \pmod p\right) \pmod q \\ &= \left(\left(s^d (rq + m) \pmod p\right) \times s^{-d} \pmod p\right) \pmod q \\ &= (rq + m) \pmod q \\ &= m. \end{aligned} \tag{5}$$

Notice that the correctness of $(rq + m) \pmod q = m$ requires $q \ll p$ and $|r|_2 + |q|_2 < |p|_2$.

The symmetric homomorphic encryption scheme proposed by Li et al. supports homomorphic addition and multiplication and is used to construct their secure outsourced comparison scheme and privacy-preserving mining solutions. Though our attack needs no homomorphic properties, we still list a brief proof, for the reason that it implies the setting of parameters.

(i) *Homomorphic addition*: For the ciphertext c_1, c_2 of two plaintexts m_1, m_2, we have

$$\begin{aligned} (c_1 + c_2) \pmod p &= s^{d_1} (r_1 q + m_1) \pmod p \\ &\quad + s^{d_2} (r_2 q + m_2) \pmod p \\ &= s^{d_1} ((r_1 + r_2) q + m_1 + m_2) \pmod p, \ d_1 = d_2. \end{aligned} \tag{6}$$

The correct decryption of $c_1 + c_2$ requires $d_1 = d_2$ and $(r_1 + r_2)q + m_1 + m_2 < p$.

(ii) *Homomorphic multiplication*: For the ciphertext c_1, c_2 of two plaintexts m_1, m_2, we have

$$\begin{aligned} (c_1 \times c_2) \pmod p &= s^{d_1} (r_1 q + m_1) \pmod p \\ &\quad \times s^{d_2} (r_2 q + m_2) \pmod p \\ &= s^{d_1 + d_2} \left(r_1 r_2 q^2 + r_1 q m_2 + r_2 q m_1 + m_1 \times m_2\right) \pmod p \\ &= s^{d_1 + d_2} \left((r_1 r_2 q + r_1 m_2 + r_2 m_1) q + m_1 \times m_2\right) \pmod p. \end{aligned} \tag{7}$$

The correct decryption of $c_1 \times c_2$ requires $(r_1 r_2 q + r_1 m_2 + r_2 m_1)q + m_1 \times m_2 < p$.

2.3. Lattice. An m-dimension lattice $\mathscr{L}$ can be regarded as a set of all integer coefficient linear combinations of basis vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$, that is $\mathscr{L}(\mathbf{B}) = \{\mathbf{v} = \sum_{i=1}^{m} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z}, \ \mathbf{b}_i \in \mathbb{R}^n\}$. If $m = n$, we call that $\mathscr{L}$ is a full-rank lattice.

One of the most famous problems involving lattice is the shortest vector problem (SVP). Given a basis of a lattice, the goal is to find one non-zero vector, which has the shortest length $\|\mathbf{v}\|$. Some approximation algorithms are usually used for solving SVP as oracles, such as LLL and BKZ algorithms. The LLL algorithm is developed by A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz [18] in 1982. Given a basis of a lattice, the LLL algorithm outputs a reduced basis, which has a smaller size by Gram-Schmidt orthogonalization. It has various applications in cryptanalysis and other fields, such as integer programming and finding integer relations.

3. The Proposed Attack

3.1. Main Idea. Define $r_i q + m_i$ as a_i, and $c_i c_1^{-1}$ as f_i. Our attack has two stages. In the first stage, we construct a lattice $\mathscr{L}(\mathbf{F})$

Input: k ciphertexts $c_1, \cdots, c_k$
Output: private key (s^d, q)
for $i = 2$ **to** k **do**
$f_i = c_i c_1^{-1}$
end for
Set $\mathscr{L}(\mathbf{F})$ with f's
$\mathbf{f}$ = LLL_reduction($\mathscr{L}(\mathbf{F})$)
$a_1 = |(\mathbf{f})_k|$
for $i = 1$ **to** $k - 1$ **do**
$a_{i+1} = |(\mathbf{f})_i|$
end for
Set $\mathscr{L}(\mathbf{A})$ with a's
$\mathbf{a}$ = LLL_reduction($\mathscr{L}(\mathbf{A})$)
$r_1 = |(\mathbf{a})_k|$
for $i = 1$ **to** $k - 1$ **do**
$r_{i+1} = (|(\mathbf{a})_i| + r_1 * a_{i+1})/a_1$
end for
Compute $s^d = c_1 a_1^{-1} \bmod p$
Compute $q = a_1/r_1 \bmod p$
return (s^d, q)

Algorithm 1: The Attack Algorithm.

with f's and run the LLL algorithm to obtain a short vector, which contains a's. In the second stage, we construct a lattice $\mathscr{L}(\mathbf{A})$ with a's and run the LLL algorithm again to obtain a short vector which contains r's. It is obvious that the secret key (s^d, q) can be computed as $s^d = ca^{-1} \bmod p$ and $q = a/r \bmod p$. Notice that there is no need for the plaintexts m's in the attack.

3.2. Details. In this part, we give a specification of the attack in Algorithm 1. The input of the attack algorithm contains a set of ciphertexts $c = \{c_1, \ldots, c_k\}$ and the modular p of the encryption scheme without plaintexts m's. The output of the attack algorithm contains s^d and q which can be used to decrypt ciphertexts.

In the first stage, the lattice $\mathscr{L}(\mathbf{F})$ is constructed as

$$\mathbf{F} = \begin{pmatrix} p & 0 & \cdots & 0 & 0 \\ 0 & p & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & p & 0 \\ -f_2 & -f_3 & \cdots & -f_k & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_k \end{pmatrix}. \quad (8)$$

Thus a short vector $\mathbf{v} \in \mathscr{L}(\mathbf{F})$ could be expressed as

$$\begin{aligned} \mathbf{v} &= x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + \cdots + x_{k-1}\mathbf{v}_{k-1} + x_k\mathbf{v}_k \\ &= (x_1 p - x_k f_2, x_2 p - x_k f_3, \ldots, x_{k-1} p - x_k f_k, x_k). \end{aligned} \quad (9)$$

Through LLL's algorithm, we could obtain a short vector $\mathbf{f} \in \mathscr{L}(\mathbf{F})$.

Claim 1. Parameters $a_2, \ldots, a_k$ are close to $|(\mathbf{f})_1|, \ldots, |(\mathbf{f})_{k-1}|$, while a_1 is close to $|(\mathbf{f})_k|$, where $(\mathbf{f})_i$ means the i'th entry of the output vector $\mathbf{f}$.

Sketch of Proof. Suppose that $f_i a_1 = a_i + t_i p$, then we have $f_i/p - t_i/a_1 = a_i/(a_1 p)$. Because $|a| \ll |p|$, $a_i/(a_1 p) \approx 1/p < negl(\lambda)$. Hence, values of $f_2/p, \ldots, f_k/p$ are close to those of $t_2/a_1, \ldots, t_k/a_1$. Because $|x_{i-1} p - x_k f_i|$ is small, we can obtain that $x_{i-1}/x_k - f_i/p < negl(\lambda)$. Hence, values of $x_1/x_k, \ldots, x_{k-1}/x_k$ are close to those of $f_2/p, \ldots, f_k/p$. Since $t_2/a_1, \ldots, t_k/a_1$ and $x_1/x_k, \ldots, x_{k-1}/x_k$ are both close to $f_2/p, \ldots, f_k/p$, we obtain that values of $t_2/a_1, \ldots, t_k/a_1$ are close to $x_1/x_k, \ldots, x_{k-1}/x_k$. With a non-negligible probability, $x_1, \ldots, x_{k-1}$ equal $t_2, \ldots, t_k$, and x_k is equal to a_1.

As $|a_1|$ is close to $|x_k|$, we have $x_k \approx |a_1|$ or $x_k \approx -|a_1|$. Considering $|a|, |x| \ll p$, we believe that $|a| < p/2$, thus $a_1 \approx |x_k|$. Similarly, as $|x_{i-1} p - x_k f_i|$ is small, we have $|x_{i-1} p - x_k f_i| = (x_{i-1} p - x_k f_i) \bmod p = -x_k f_i \bmod p$ or $|x_{i-1} p - x_k f_i| = -(x_{i-1} p - x_k f_i) \bmod p = x_k f_i \bmod p$, i.e. $|x_{i-1} p - x_k f_i| = |x_k| f_i \bmod p$, thus $a_{i+1} = a_1 f_{i+1} \bmod p \approx |x_k| f_{i+1} \approx (\mathbf{f})_i$, $i = 1, \ldots, k-1$.

In the second stage, the lattice $\mathscr{L}(\mathbf{A})$ is constructed as

$$\mathbf{A} = \begin{pmatrix} a_1 & 0 & \cdots & 0 & 0 \\ 0 & a_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_1 & 0 \\ -a_2 & -a_3 & \cdots & -a_k & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_k \end{pmatrix}. \quad (10)$$

Thus a short vector $\mathbf{w} \in \mathscr{L}(\mathbf{A})$ could be expressed as

$$\begin{aligned} \mathbf{w} &= y_1\mathbf{w}_1 + y_2\mathbf{w}_2 + \cdots + y_{k-1}\mathbf{w}_{k-1} + y_k\mathbf{w}_k \\ &= (y_1 a_1 - y_k a_2, y_2 a_1 - y_k a_3, \ldots, y_{k-1} a_1 - y_k a_k, y_k). \end{aligned} \quad (11)$$

Through LLL's algorithm, we could obtain a short vector $\mathbf{a} \in \mathscr{L}(\mathbf{A})$.

Claim 2. Parameters $r_1, r_2, \ldots, r_k$ are close to $|(\mathbf{a})_k|, (|(\mathbf{a})_1| + r_1 * a_2)/a_1, \ldots, (|(\mathbf{a})_{k-1}| + r_1 * a_k)/a_1$, where $(\mathbf{a})_i$ means the i'th entry of the output vector $\mathbf{a}$.

Sketch of Proof. Considering $a_i = r_i q + m_i$, we have $a_i/a_1 = (r_i q + m_i)/(r_1 q + m_1) \approx (r_i q)/(r_1 q) = r_i/r_1$. Hence, values of $a_2/a_1, \ldots, a_k/a_1$ are close to those of $r_2/r_1, \ldots, r_k/r_1$.

Because $|y_{i-1} a_1 - y_k a_i|$ is small, we can obtain that $y_{i-1}/y_k - a_i/a_1 < negl(\lambda)$. Hence, values of $y_1/y_k, \ldots, y_{k-1}/y_k$ are close to those of $a_2/a_1, \ldots, a_k/a_1$. Since $r_2/r_1, \ldots, r_k/r_1$ and $y_1/y_k, \ldots, y_{k-1}/y_k$ are both close to $a_2/a_1, \ldots, a_k/a_1$, we obtain that values of $r_2/r_1, \ldots, r_k/r_1$ are close to $y_1/y_k, \ldots, y_{k-1}/y_k$. With a non-negligible probability, $y_1, \ldots, y_{k-1}$ equal $r_2, \ldots, r_k$, and y_k is equal to r_1.

Likewise, we have $r_1 = |(\mathbf{a})_k|$ and $r_{i+1} = (|(\mathbf{a})_i| + r_1 * a_{i+1})/a_1$, $i = 1, \ldots, k-1$.

Since we have recovered all the a's and r's, the secret key (s^d, q) could be simply computed as $s^d = c_i a_i^{-1}$ and $q = a_i/r_i$. Parameters c_1, a_1 and r_1 are used to compute (s^d, q) in the algorithm.

Table 1: Results in experiments with different parameters.

$\|p\|_2$	$\|q\|_2$	$\|r\|_2$	m	k	Instances	Successes	Average Time
256	64	64	$\leq q/256$	20	100	98	28.65s
256	64	64	$\leq q/64$	20	100	10	29.20s
256	64	96	$\leq q/256$	20	100	23	39.70s
256	64	48	$\leq q/256$	20	100	2	25.09s
256	48	48	$\leq q/32$	20	100	51	20.56s

Table 2: Example for $|p|_2 = 256$.

The Encryption Scheme	KeyGen	p	96676600900811618538103427778952879986194318084870939447026698628675235799451
		q	17957200991146257161
		s	5710889004555322387
	E()	d	1
		m_1	47863226783593508
		r_1	13683909070104700313
		c_1	1403306518881428241485832704008411265962802817946162878687
		...	...
		...	...
		...	...
		m_{20}	8614814073974658
		r_{20}	15727429030749794270
		c_{20}	1612872723066650760854493964947953343890747803002904919536
The Cryptanalytic Algorithm	The First Stage	v_1	-19946395763615474623999402780809496189 3
		...	...
		v_{20}	24572470551643938262638513685031878 4901
	The Second Stage	w_1	-23371144323647625378158005673135813 3
		...	...
		w_{20}	13683909070104700313
		s	5710889004555322387
		q	17957200991146257161

3.3. Experiments. We run our proposed cryptanalytic algorithm on a personal computer using NTL library [19]. The environment is listed as follows:

(i) CPU: Intel(R) Core (TM) i3-7100 3.90GHz

(ii) RAM: 4.00GB

(iii) OS: Windows 10 64bit

Notice that the output of the attack algorithm is s^d. In [16], the parameter d is called *ciphertext degree* and is believed to be a small positive integer. It means that we could collect enough *d-degree ciphertexts* we need, and it is not difficult to recover s from s^d. For the *d-degree ciphertexts*, the encryption and decryption algorithms only require s^d rather than s. Thus, it is sufficient to break the scheme if we can recover s^d. For convenience, we suppose the parameter $d = 1$ in the encryption algorithm. When $d \neq 1$, our algorithm still works correctly.

The results are given in Table 1. As a result of the approximation, the chance of success is relavant to $|q|_2$, $|r|_2$, and $|m|_2$. The best situation is when $|q|_2 \approx |r|_2$ and $|m| \ll |rq|$. To make it easier to understand our proposed attack, we give an example to illustrate the procedure of the algorithm in Table 2. The parameters are set as $|p|_2 = 256$, $|q|_2 = 64$, $|r|_2 = 64$, and $m \leq q/256$.

Firstly, we compute all the f's with the input ciphertexts $(c_1, \ldots, c_k)$. Secondly, we use LLL algorithm to obtain a short vector for solving all a's. Thirdly, we use LLL algorithm again to obtain a short vector for solving all r's. Finally, we compute secret key (s, q) with a's and r's.

In practice, the first row $\mathbf{f}$ ($\mathbf{a}$, respectively) of the reduced basis of $\mathscr{L}(\mathbf{F})$ ($\mathscr{L}(\mathbf{A})$, respectively) which is a row vector with a short norm; thus we regard it as the short vector $\mathbf{v}$ ($\mathbf{w}$, respectively) we need.

The chance of success depends on the bit lengths of r, q, and m. In the first stage, $|a| \ll |p|$ requires $|rq| \ll |p|$. In the second stage, $a_i/a_1 = (r_i q + m_i)/(r_1 q + m_1) \approx (r_i q)/(r_1 q) = r_i/r_1$ requires $|m| \ll |rq|$. Thus we need to hold $|m| \ll |rq| \ll |p|$. Besides, the recovery of q from a also limit the setting of parameters. Notice that $a = rq + m$, where $0 \leq m < q$. If $m <$

Table 3: Comparision of Wang et al.'s Attack and Ours.

	$\|p\|_2$	$\|q\|_2$	$\|r\|_2$	Chance of Success	Average Time	Plaintexts Needed?
Attack in [17]	241	80	40	98%	0.1292s	Yes
Our Attack	256	64	64	98%	28.65s	No

r, the result of a/r is equal to q. However, when $|r|_2 < |q|_2$, we cannot confirm that $0 \le m < r$. In conclusion, the best situation is when $|r| \approx |q|$ and $|r|$ is slightly greater than $|q|$.

3.4. Complexity Analysis. We start with some simple conclusions about computational complexity.

(1) The computational complexity of modular inverse modulo p is $O(\log^3 p)$.

(2) The computational complexity of modular multiplication modulo p is $O(\log^2 p)$.

(3) The computational complexity of the LLL algorithm is $O(N^5(\log^2 B))$ [20], where N is the dimension of the lattice, and B is the maximum length of input basis under the Euclidean norm.

Combining (1) and (2), we can conclude that the computational complexity of calculating f's, r's, s, and q is $O(\log^3 p)$. In our attack, $N = k$ and $B \le \sqrt{(k-1)p^2 + 1}$. We can obtain the computational complexity of the LLL algorithm

$$\begin{aligned} O\left(N^5\left(\log^2 B\right)\right) &= O\left(k^5 \log^2 \sqrt{(k-1)\,p^2 + 1}\right) \\ &= O\left(k^5 \log^2 \sqrt{kp^2}\right) = O\left(k^5\left(\frac{1}{2}\log k + \log p\right)^2\right) \\ &= O\left(\frac{1}{4}k^5 \log^2 k + k^5 \log k \log p + k^5 \log^2 p\right). \end{aligned} \tag{12}$$

Because $k < p$, we can obtain $\log k < \log p$, thus $O(N^5(\log^2 B)) = O(k^5 \log^2 p)$.

In practice, the computational complexity of our attack is mainly dependent on that of the LLL algorithm. For example, suppose that $k^5 \log^2 p \le \log^3 p$ we can obtain $\log p \ge k^5$. If we set $k = 20$, then $\log p \ge 20^5 \approx 2^{21}$. It means that p is a 2^{21}-bit-length prime, while the bit length of the prime we usually use is $2048 = 2^{11}$ or $4096 = 2^{12}$.

Above all, the computational complexity of our attack algorithm is $O(k^5 \log^2 p)$. Obviously, it is worse than the complexity $O(\log^4 p)$ of Wang et al.'s attack [17]; however, our attack eliminates the assumption of known plaintext/ciphertext pairs.

3.5. Discussions. Notice that in the attack algorithm, the output of the LLL algorithm is a vector, such as $\mathbf{f}$ and $\mathbf{a}$, rather than a reduced basis. We regard the first row vector of an LLL-reduced basis as the goal short vector. We explain the reason below.

An δ-LLL-reduced $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ has two important properties:

(1) $\forall 1 \le i \le n,\ j < i,\ |\mu_{i,j}| \le 1/2$,

(2) $\forall 1 \le i \le n,\ \delta\|\tilde{\mathbf{b}}_i\|^2 \le \mu_{i+1,i}^2\|\tilde{\mathbf{b}}_i\|^2 + \|\tilde{\mathbf{b}}_{i+1}\|^2$,

where $\tilde{\mathbf{B}} = \{\tilde{\mathbf{b}}_1, \tilde{\mathbf{b}}_2, \ldots, \tilde{\mathbf{b}}_n\}$ is the Gram-Schmidt orthogonalization of $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, and the coefficient $\mu_{i,j} = \langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle / \langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle$.

From these two properties, we can conclude that

$$\left\|\tilde{\mathbf{b}}_1\right\| \le 2^{(n-1)/2} \lambda_1(\mathscr{L}), \tag{13}$$

where $\lambda_1(\mathscr{L})$ is the length of the shortest non-zero vector in $\mathscr{L}(\mathbf{B})$. Please refer to [21] for more detailed introduction and proof.

The efficiency of our attack algorithm is mainly subject to the parameter k. Smaller k implies a greater chance of success for the reason that a's and r's can be recovered from the LLL algorithm easier while the runtime of the LLL algorithm rises rapidly. In our experiment, we recommend that k should be set as 20 considering both the chance of success and the runtime. In addition, the chance of success is also limited by sizes of r, q, and m.

Table 3 gives a comparision of Wang et al's attack and ours. Although the bit lengths of the parameters $|p|_2$, $|q|_2$ and $|r|_2$ are close but different, the average time of Wang et al's is much less than ours. However, the improved attack algorithm eliminates the assumption of known plaintext/ciphertext pairs, thus a ciphertext-only adversary can break the encryption scheme through this way.

4. Conclusion

In this paper, we propose a new attack algorithm on the symmetric homomorphic encryption scheme presented by Li et al. Our attack can recover the secret key pair from several ciphertexts without plaintexts. In our experiment, the attack an be finished during an acceptable period of time with recovering most of the secret key in the generated instances. For the cases $|p|_2 = 256$, $|q|_2 = |r|_2 = 64$, $m \le q/256$, the key-recovery cryptanalytic algorithm only takes about 29 seconds. Although the running time and the opportunity of success depend on the sizes of parameters, the attack algorithm can still be used in real practice to recover secret key pairs.

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 are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant No. 2017YFB0802000, the National Natural Science Foundation of China under Grant Nos. 61572390, U1736111, the National Cryptography Development Fund under Grant No. MMJJ20180111, the Plan For Scientific Innovation Talent of Henan Province under Grand no. 184100510012, the Program for Science & Technology Innovation Talents in Universities of Henan Province under Grant No. 18HASTIT022, and the Innovation Scientists and Technicians Troop Construction Projects of Henan Province, Science & technology planning project in Henan Province (182102210124).

References

[1] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[2] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[3] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[4] M. Zhang, Y. Yao, Y. Jiang, B. Li, and C. Tang, "Accountable mobile E-commerce scheme in intelligent cloud system transactions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1889–1899, 2018.

[5] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, pp. 169–179, 1978.

[6] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology —EUROCRYPT '99*, J. Stern, Ed., vol. 1592, pp. 223–238, Springer, Berlin, Germany, 1999.

[7] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of Computing (STOC '09)*, pp. 169–178, ACM, New York, NY, USA, 2009.

[8] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, pp. 97–106, Palm Springs, Calif, USA, October 2011.

[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Advances in Cryptology – CRYPTO 2011*, R. Phillip, Ed., vol. 6841, pp. 505–524, Springer, Berlin, Germany, 2011.

[10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in cryptology—EUROCRYPT 2010*, H. Gilbert, Ed., vol. 6110, pp. 24–43, Springer, Berlin, Germany, 2010.

[11] F. Armknecht and T. Strufe, "An efficient distributed privacy-preserving recommendation system," in *Proceedings of the 2011 the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net'2011*, pp. 65–70, Italy, June 2011.

[12] C. Bosch, A. Peter, P. Hartel, and W. Jonker, "SOFIR: Securely outsourced Forensic image recognition," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2014*, pp. 2694–2698, Italy, May 2014.

[13] A. Jeckmans, A. Peter, and P. Hartel, "Efficient privacy-enhanced familiarity-based recommender system," in *Computer Security – ESORICS 2013*, J. Crampton, S. Jajodia, and K. Mayes, Eds., vol. 8134 of *Lecture Notes in Computer Science*, pp. 400–417, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[14] M. Naehrig, L. Kristin, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the ACM Cloud Computing Security Workshop, Ccsw 2011*, pp. 113–124, Chicago, Ill, Usa, October 2011.

[15] Z. Yang, S. Zhong, and R. N. Wright, "Privacy-preserving classification of customer data without loss of accuracy," in *Proceedings of the SDM*, pp. 92–102, 2005.

[16] L. Li, R. Lu, K.-K. R. Choo, A. Datta, and J. Shao, "Privacy-preserving-outsourced association rule mining on vertically partitioned databases," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1547–1861, 2016.

[17] B. Wang, Y. Zhan, and Z. Zhang, "Cryptanalysis of a symmetric fully homomorphic encryption scheme," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1460–1467, 2018.

[18] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[19] V. Shoup, *Ntl: A Library for Doing Number Theory*, vol. 01, 2003.

[20] Y. Park and J. Park, "Analysis of the upper bound on the complexity of LLL algorithm," *Journal of the Korean Society for Industrial and Applied Mathematics*, vol. 20, no. 2, pp. 107–121, 2016.

[21] O. Regev, "Lattices in computer science," in *Proceedings of the Lecture notes of a course given in Tel Aviv University*, vol. 31, 2004.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 2516963, 13 pages
https://doi.org/10.1155/2019/2516963

Review Article

Revisiting Anonymous Two-Factor Authentication Schemes for IoT-Enabled Devices in Cloud Computing Environments

Ping Wang,[1,2,3] **Bin Li**,[1] **Hongjin Shi**,[4] **Yaosheng Shen**,[2] **and Ding Wang**[4]

[1]*School of Software and Microelectronics, Peking University, Beijing 100871, China*
[2]*School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China*
[3]*National Engineering Research Center for Software Engineering, Beijing, China*
[4]*School of EECS, Peking University, Beijing 100871, China*

Correspondence should be addressed to Ding Wang; wangdingg@pku.edu.cn

Received 2 February 2019; Accepted 7 May 2019; Published 23 May 2019

Guest Editor: Fagen Li

Investigating the security pitfalls of cryptographic protocols is crucial to understand how to improve security. At ICCCS'17, Wu and Xu proposed an efficient smart-card-based password authentication scheme for cloud computing environments to cope with the vulnerabilities in Jiang et al.'s scheme. However, we reveal that Wu-Xu's scheme actually is subject to various security flaws, such as offline password guessing attack and replay attack. Besides security, user friendly is also another great concern. In 2017, Roy et al. found that in most previous two-factor schemes a user has to manage different credentials for different services and further suggested a user-friendly scheme which is claimed to be suitable for multiserver architecture and robust against various attacks. In this work, we show that Roy et al.'s scheme fails to achieve truly two-factor security and shows poor scalability. At FGCS'18, Amin et al. pointed out that most of existing two-factor schemes are either insecure or inefficient for mobile devices due to the use of public-key techniques and thus suggested an improved protocol by using only light-weight symmetric key techniques. Almost at the same time, Wei et al. also observed this issue and proposed a new scheme based on symmetric key techniques with formal security proofs in the random oracle model. Nevertheless, we point out that both Amin et al.'s and Wei et al.'s schemes cannot achieve the claimed security goals (including the most crucial goal of "truly two-factor security"). Our results invalidate any use of the scrutinized schemes for cloud computing environments.

1. Introduction

With the emerging paradigm of cloud computing and Internet of Things (IoT), various services are provided over the cloud and accessed by users from IoT-enable devices (e.g., mobile phones and smart watches). As cloud-based services can be accessed anytime and anywhere just with a connection to the Internet and IoT-enable devices are generally of resource-constrained nature, it is important and challenging to protect users and cloud servers from severe security threats, such as fraudulence, eavesdropping, and falsification, posed either by external attackers or malicious internal entities. To guarantee that the resources and services can only be accessed by legitimate parties, user authentication plays an important part in securing electronic transactions by acquiring collaborative evidence. In 2011, Hao et al. [1] suggested the first two-factor authentication scheme that is based on password and smart cards for the cloud computing environments. This initial work has brought about a number of enhanced proposals [2–9] with each different in terms of security [10], anonymity [11], usability [12], and efficiency [13].

Without loss of generality, we consider the most common client-server architecture in which two participants (i.e., a server S and a user U) get involved in two-factor authentication [14]. User U holds a memorable password and a smart card stored with some initial security parameters, and server S only needs to keep some secret key material of the system. Since there is no need to keep a table with password-related verification information on the server side, the server is free from the threat of password dataset leaks and ameliorated from the burden of maintaining a large password dataset.

This feature that makes this type of schemes rather desirable, considering that there are incessant leakages of password databases from large websites[15]. The most important security goal of this kind of schemes is the so-called "two-factor security" [16]. This security concept essentially means that only the user who has the smart card and knows the correct password can be verified by the server.

Most of existing two-factor schemes (e.g, [5, 8, 23]) for cloud computing are built on the basis of generic two-factor schemes like [3, 24]. Nevertheless, past research [20, 25–27] have, again and again, proved that designing a smart-card-based password authentication scheme that can attain "two-factor security" is a tough task. In 2009, Xu et al. [28] developed a smart-card-based password authentication scheme relying on the intractability of computational Diffie-Hellman problem, and stated that their scheme is able to support "two-factor security" under the hypothesis that smart cards can be tampered. In addition, their scheme was "proved secure" in the random oracle model. However, later on Sood et al. [29] found that Xu et al.'s scheme cannot resist against user impersonation attack if the parameters kept in the smart cards can be extracted, invalidating Xu et al.'s claim of ensuring "two-factor security". In 2010, Song [30] independently found this severe flaw in Xu et al.'s scheme. Furthermore, Song presented an improvement to counter the problem emerged in Xu et al.'s scheme.

In 2012, Chen et al. [31] pointed out that various security drawbacks still existed in both Sood et al.'s [29] and Song's [30] schemes. More specifically, Sood et al.'s scheme is susceptible to server impersonation attack and Song's scheme suffers from offline password guessing attack, in case the attacker can obtain those sensitive information kept in the smart card. Chen *et al.* [31] also put forward an improved scheme and argued that their scheme is robust under the condition that the sensitive data in smart card has been revealed by the attacker. It should be noted that recent rapid developments in side-channel attacks have proved that the sensitive information stored in general commercial smart cards could be extracted by power analysis or reverse engineering [32, 33]. Based on a weak yet realistic assumption, Chen et al.'s scheme [31] appears very practical.

However, soon after Chen et al.'s scheme [31] was presented, Ma et al. [20] figured out that it is susceptible to exactly the same problem (i.e., prone to offline password guessing attack and no supply of forward secrecy) with the original scheme (i.e., Song's scheme [30]). Based on their past experience of protocol design and analysis, for the first time Ma et al. [20] suggested three generic principles for designing a secure and efficient two-factor protocol, namely, the public-key principle, the forward secrecy principle, and the security-usability balance principle. Unfortunately, none of the two-factor authentication protocols mentioned above can satisfy all these three design principles and, moreover, as we illustrate, all the schemes studied in this work fail to comply with at least one of these principles.

In 2017, Wu and Xu [17] also observed that previous schemes (e.g., [30, 31]) are vulnerable to various security loopholes (e.g., user impersonation attack and insider attack) and also suggested an enhanced scheme. Wu-Xu argued that their new scheme not only eradicates the security pitfalls being overlooked in previous schemes but also maintains strengths of previous schemes. Notwithstanding their claims, we will show that this scheme still has several serious defects being overlooked: (1) it cannot withstand offline password guessing attack if the data in smart card can be revealed, which means that the primary goal of "truly two-factor security" cannot be satisfying; (2) it is subject to replay attack; (3) it ensures no timely typo detection.

Later on, Roy et al. [19] found that, in Tsai-Lo's scheme [34], a user has to manage different credentials for different services and further suggested a user-friendly scheme which is claimed to be suitable for multiserver architecture and robust against various attacks. Therefore, this protocol shows a good application potential in multiserver cloud computing environments. In this work, we show that Roy et al.'s scheme fails to achieve truly two-factor security and cannot preserve user untraceability. We observe that the first failure of Roy et al.'s scheme is due in large part to the noncompliance with Ma et al.'s [20] security-usability tradeoff principle. Our attacks highlight the necessity of being aware of basic protocol design principles.

In 2018, Amin et al. [21] pointed out that previous schemes (e.g., [35–37]) are vulnerable to various security loopholes (e.g., off-line password guessing attack, insider attack and user impersonation attack) and also developed a new scheme for distributed cloud computing environments. Amin et al. argued that their new scheme not only eradicates the security pitfalls being overlooked in previous schemes but also maintains strengths of previous schemes. Notwithstanding their claims, we will show that this scheme still has several serious defects being overlooked: (1) it cannot withstand offline password guessing attack if the data in smart card can be revealed, which means that the primary goal of "truly two-factor security" cannot be satisfied with; (2) it provides no forward secrecy; (3) it ensures no user untraceability.

Very recently, Wei et al. [23] observed that most of previous schemes (e.g., [36–40]) only provided some heuristic security arguments and little attention has been paid to the formal treatment of protocol security. Unsurprisingly, most of them are vulnerable to various security loopholes. Thus, they suggested an enhanced scheme for cloud computing without using computation-expensive public key operations. They employed a new cryptographic primitive called authenticated encryption scheme which can guarantee both message confidentiality and integrity, and showed that their new scheme can be provably secure in the random oracle model. Though this scheme is armed with a formal proof, we will show that this scheme still has several serious defects being overlooked: (1) it cannot achieve the primary goal of "truly two-factor security"; (2) it provides no forward secrecy; (3) it ensures no user untraceability.

The remainder of the paper is organized as follows: we revisit Wu-Xu's scheme in Section 2; we describe the security loopholes of Roy et al.'s scheme in Section 3; Amin et al.'s scheme is scrutinized in Section 4 and Wei et al.'s scheme is cryptanalyzed in Section 5. Finally, we conclude the paper in Section 6.

Table 1: Notations and abbreviations.

Symbol	Description
U_i	i^{th} user
S	remote server
$\mathcal{M}$	malicious attacker
ID_i	identity of user U_i
PW_i	password of user U_i
x	the secret key of remote server S
$\oplus$	the bitwise XOR operation
$\|$	the string concatenation operation
$h(\cdot)$	collision free one-way hash function
$\longrightarrow$	a common channel
$\Longrightarrow$	a secure channel

2. Revisiting Wu-Xu's Scheme

2.1. Review Wu-Xu's Scheme. In this section, we briefly review the chaotic-map based authentication scheme for cloud computing proposed by Wu and Xu [17] in ICCCS 2017. Their scheme consists of four phases: initialization, registration, authentication, and password change. For ease of presentation, we will follow the notations in Wu-Xu's scheme as closely as possible and summarize the notations in Table 1.

2.1.1. Registration Phase. At the very start, the server S generates a random positive integer $s \in Z_p^*$ and a symmetric key cryptosystem with $E_k(.)$ and $D_k(.)$ and then selects a secret key x, and two one-way hash functions $h(.)$ and $h_1(.)$.

Step R1. U_i chooses her identity ID_i, PW_i and b_i, then computes $HPW_i = h(PW_i \| b_i)$

Step R2. $U_i \Longrightarrow S : \{ID_i, HPW_i\}$.

Step R3. S selects a random number r_i, computes $IM_i = E_x((ID_i \oplus r_i) \| r_i)$ and $B_1 = h(ID_i \| HPW_i) \oplus HPW_i \oplus h(x \| IM_i)$, and stores IM_i, B_1, $h(.)$, $E_k(.)/D_k(.)$, s and p into the smart card.

Step R4. $S \Longrightarrow U_i$: A smart card containing $\{IM_i, B_1, E_k(.)/D_k(.), h(.), s, p\}$.

Step R5. U_i computes $B_2 = h(ID_i \| PW_i) \oplus b_i$ and stores B_2 into the card.

2.1.2. Login and Mutual Authentication Phase. When U_i wants to login, the following operations will be performed:

Step 1. U_i inserts the smart card into card reader and inputs ID_i and PW_i.

Step 2. Smart card computes $b_i' = B_2 \oplus h(ID_i \| PW_i)$.

Step 3. Smart card selects two random integers $u \in [1, p + 1]$ and r_u and computes $HPW_i = h(PW_i \| b_i')$, $C_1 = T_u(s)$, $C_2 = B_1 \oplus h(ID_i \| HPW_i)$, $C_3 = h(ID_i \| C_1 \| C_2 \| r_u)$, $C_4 = C_2 \oplus C_3$, $C_5 = E_{C_3}(ID_i \| C_1 \| r_u)$.

Step 4. Smart card$\longrightarrow S : \{C_4, IM_i, C_5\}$.

Step 5. On receiving the message from smart card, S decrypts IM_i and gets ID_i' and r_i' and then computes $C_2' = h(x \| IM_i)$, $C_3' = C_4 \oplus C_2'$.

Step 6. S decrypts C_5 to obtain ID_i'', C_1', and r_u', then checks if $ID_i' \stackrel{?}{=} ID_i''$ and $C_3' \stackrel{?}{=} h(ID_i' \| C_1' \| C_2' \| r_u')$. If all above verifications are correct, S proceeds. Otherwise, the login request is interrupted.

Step 7. S chooses three random integers $v \in [1, p + 1]$, r_s and r_i^{new} and computes $IM_i' = E_x((ID_i' \oplus r_i^{new}) \| r_i^{new})$, $C_6 = T_v(s)$, $sk_s = T_v(C_1')$, $C_7 = h(x \| IM_i')$, $C_8 = h_1(ID_i' \| IM_i' \| C_1' \| C_2' \| C_6 \| C_7 \| sk_s \| r_s)$, $C_9 = C_2' \oplus C_8$, and $C_{10} = E_{C_8}(IM_i' \| C_6 \| C_7 \| r_s)$, where sk_s is server-side session key.

Step 8. $S \longrightarrow U_i : \{C_9, C_{10}\}$.

Step 9. Smart card calculates $C_8' = C_9 \oplus C_2$ and obtains IM_i'', C_6', C_7', and r_s' by decrypting C_{10}. Smart card further computes $sk_u = T_u(C_6')$ and checks $C_8' \stackrel{?}{=} h_1(ID_i \| IM_i'' \| C_1 \| C_2 \| C_6' \| C_7' \| sk_u \| r_s')$. If they are not equal, U_i aborts the communication. Otherwise, smart card computes $B_1' = C_2 \oplus C_7' \oplus B_1$ and replaces (B_1, IM_i) with (B_1', IM_i'').

2.2. Cryptanalysis of Wu-Xu's Scheme. In this section, the security loopholes of Wu-Xu's scheme [17] will be pointed out. More specifically, it is prone to offline password guessing attack and suffers from replay attack, which makes this scheme unpractical for real use. Before giving the detailed security analysis, we first define the various adversary models for smart-card-based password authentication.

2.2.1. Adversary Models. To analyze the security provisions of password-based authentication schemes using smart cards, generally three assumptions about the attacker's capabilities are made since the landmark work of Yang et al. [24], and we summarize them as follows:

Assumption 1. The malicious attacker $\mathcal{M}$ is able to eavesdrop, delete, insert, modify, or block any transcripts communicated in the public channel. That is to say, the communication channel between the common users and the server can be completely manipulated by $\mathcal{M}$. This well complies with the standard adversary model that is widely accepted for distributed computing [41].

Assumption 2. The malicious attacker $\mathcal{M}$ can somehow get the victim user's smart card and use side-channel attack techniques to acquire sensitive security parameters from the card memory, which is reasonable according to the recent research developments in side-channel attacks [32, 33].

Assumption 3. User's password space is very constrained and the malicious attacker $\mathcal{M}$ can offline enumerate it. To be user-friendly, most protocols (e.g., the ones in [3, 24, 42–44]) enable the users to select their own password at will in the initial process of registration or later process of password change. Because human beings are incapable of memorizing random strings, instead they are likely to choose passwords that relate to their personal lives or short strings for convenience. As a result, these human-generated passwords are often very weak and belong to a small dictionary [45–47].

Note that the nontamper resistant assumption about smart cards are conditional, i.e., under the conditions that (1) smart cards have somehow been in the possession of the attacker for a relatively long time (e.g., at least a few days), which should be sufficient for launching a side-channel attack; (2) the user is not on the scene; otherwise the user will observe the attack, for side-channel attacks generally need special instruments and professional/particular operations. This is why smart cards are superior to memory sticks, even though nontamper assumption about smart cards has been made: memory sticks are nontamper resistant unconditional. See more explications in [48].

Obviously, if both *Assumptions* 2 and 3 hold simultaneously, then an attacker (without any other assumptions/abilities) can successfully impersonate a victim user and any scheme is trivially insecure. Therefore, it is widely regarded that attackers should not be granted to obtain a victim user's smart card as well as his password [3, 24, 49].

In [17], Wu and Xu reported that their scheme is secure under the above three assumptions. For example, they stated that their scheme can withstand offline password guessing attacks even if the security parameters in smart card have been extracted. However, contrary to their claims, we will illustrate that this scheme is still vulnerable to offline guessing as well as other pitfalls. Based on the above listed assumptions, we cryptanalyze the security provisions of Wu-Xu's scheme in the following and assume that $\mathcal{M}$ can extract the secret data $\{B_1, B_2, h(\cdot), E_k(\cdot), D_k(\cdot)\}$ stored in U_i's smart card, and $\mathcal{M}$ can also eavesdrop the messages $\{C_4, IM_i, C_5\}$ exchanged between the parties.

2.2.2. Offline Password Guessing Attack. It is known that password-based authentication schemes are apt to be subject to two kinds of attacks regarding guessing [16, 25], i.e., offline password guessing and online password guessing, due to the limited size of the password space. Among them, the online guessing is relatively easy to detect due to the abnormal, large number of login requests issued by the attacker against the victim account within a short duration, and thus it can be countered by rate-limiting [50].

It is worth noting that when targeted online password guessing [47] is considered, attackers will no longer need to launch a large number of login attempts to try candidate passwords. Instead, targeted online password guessing attacks are rather effective and, as reported by Wang et al. [47], the attacker $\mathcal{M}$ can have an over 20% of success rate within 100 login attempts against normal users, if $\mathcal{M}$ has obtained some personal information (e.g., name, birthday) of the victim user. As a quick response to the results of [47], the US Digital identity guideline NIST 800-63B [51] has been revised and the following sentence was removed: "online guessing can be readily addressed by throttling the rate of login attempts permitted" (see more details in [52, 53]). Further, NIST 800-63B proposed four countermeasures such as CAPTCHA, exponential time delay, and risk-based authentication (see Section 5.2.2 of [51]).

In contrast, offline password guessing attack cannot be easily detected. In this attack, the attacker $\mathcal{M}$ first attempts to look for some pieces of information that can be exploited as the comparison target of his password guesses and then locally determines the exactly correct password by repeatedly testing all the candidates. Since this attack is executed without online communication with the server, there is no means for the server to detect and thwart. In all, no matter in trawling or targeted online guessing, $\mathcal{M}$ needs to interact with the server and there is the possibility that $\mathcal{M}$ may be detected, but in offline password guessing there is no such possibility. Consequently, offline password guessing is a more serious threat if password-related information has been leaked.

Wu and Xu [17] claimed that an attacker is unable to, in an offline manner, determine a user's password even if the sensitive data B_1 has been revealed from user's smart card. However, the following attacking procedure illustrates that this claim is not tenable. Suppose that U_i's smart card is lost/stolen and the attacker $\mathcal{M}$ obtains it. Then $\mathcal{M}$ extracts the content $\{B_1, B_2, h(\cdot)\}$ by using the methods introduced in [33]. The following procedure describes our proposed offline password guessing attack:

Step 1. $\mathcal{M}$ choose a pair (ID_i^*, PW_i^*) from the identity space $\mathcal{D}_{ID}$ and dictionary space $\mathcal{D}_{PW}$, respectively.

Step 2. $\mathcal{M}$ computes $b_i^* = B_2 \oplus h(ID_i^* \parallel PW_i^*)$, where B_2 is revealed from U_i's smart card.

Step 3. $\mathcal{M}$ computes $C_2^* = B_1 \oplus h(ID_i^* \parallel h(PW_i^* \parallel b_i^*)) \oplus h(PW_i^* \parallel b_i^*)$, where B_1 is recovered from U_i's smart card.

Step 4. $\mathcal{M}$ calculates $C_3^* = C_2^* \oplus C_4$, where C_4 is eavesdropped from the open channel.

Step 5. $\mathcal{M}$ decrypts C_5 by using C_3^* to obtain $ID_i' \oplus C_1' \oplus r_u'$, where C_5 is eavesdropped from the open channel.

Step 6. $\mathcal{M}$ calculates $C_3^* = h(ID_i^* \parallel C_1' \parallel C_2^* \parallel r_u')$.

Step 7. $\mathcal{M}$ examines the authenticity of (ID_i^*, PW_i^*) pair by verifying if $C_4 \stackrel{?}{=} C_2^* \oplus C_3^*$.

Step 8. $\mathcal{M}$ returns to Step 1 of this procedure until the right pair of (ID_i, PW_i) is obtained or all pairs in $\mathcal{D}_{ID} \times \mathcal{D}_{PW}$ are exhausted.

It is obvious that the above procedure is with a time complexity of $\{\mathcal{O}(|\mathcal{D}_{ID}| * |\mathcal{D}_{PW}|)\} * (T_D + 4T_H + 7T_X)$, where T_E, T_H, and T_X denote the execution time of modular exponentiation, hash, and XOR operation, respectively. Based on the results reported in [16, 54], the offline password

guessing attack is able to be carried out in seconds on a common computer, for in practice the size of identity space $\mathscr{D}_{ID}$ and the size of dictionary space $\mathscr{D}_{PW}$ are rather limited and $\mathscr{M}$ could try all the possible passwords through an offline method [20, 49]. All in all, an type II attacker $\mathscr{M}$ can guess (ID_i, PW_i) within polynomial time bound; it follows that our suggested attack is indeed effective.

2.2.3. Replay Attack. Resistance to replay attack is a very basic security goal of any cryptographic protocol [16, 24]. However, Wu-Xu's scheme fails to achieve this goal. More specifically, Wu-Xu's scheme employs random numbers but not timestamps to achieve the freshness of messages. Yet, this scheme has only two protocol flows, making it inherently vulnerable to replay attack. As is well known, any two-flow random number based scheme is unable to achieve explicit authentication while resisting replay attack, because $\mathscr{M}$ can simply replay the first message of a successful protocol run to impersonate the legitimate user, and the server can never know whether the replayed message is fresh or not unless the server maintains a table of all received messages. However, maintaining a table of all received messages is practically undesirable. In all, replay attack is quite realistic against Wu-Xu's scheme.

3. Revisiting of Roy Et Al.'s Scheme

3.1. Review of Roy Et Al.'s Scheme. In this section, we first concisely review Roy et al.'s scheme [19] proposed in 2017. This scheme is an improvement over Tsai-Lo's scheme [34] and aims to attain forward secrecy that is lacked in Tsai-Lo's scheme. Roy et al.'s protocol involves three participants, i.e., the mobile user (MU_i), the cloud server (CS_j), and registration center (RC). There are five phases in their scheme: registration, login, authentication and key establishment, password change, and revocation of mobile device. The notations and initial system parameters employed in Roy et al.'s scheme are same as employed in the scheme of Wu-Xu (see Table 1).

3.1.1. Mobile User Registration

Step 1. MU_i chooses her identity ID_i, password PW_i, biometrics B_i, and two 128-bit random numbers b and k.

Step 2. MU_i produces $(\theta_i, \phi_i) = Gen(B_i)$ and computes the masked password $RPWB_i = H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b))$.

Step 3. $MU_i \Longrightarrow RC : \{ID_i, (RPWB_i \oplus k)\}$.

Step 4. RC selects an 1024-bit master secret key X_j for server CS_j. RC also selects an 1024-bit random number r_{ij} for each MU_i and CS_j pair.

Step 5. RC computes $A_{ij} = H(H(ID_i \oplus r_{ij}) \parallel X_j)$, $V_{ij} = A_{ij} \oplus RPWB_i$, and $RID_{S_j} = H(ID_{S_j} \parallel X_j)$ as the pseudoidentity of CS_j.

Step 6. RC selects a unique and random temporary identity TID_i for MU_i and saves n server key-plus-id combinations $\{TID_i, (ID_{S_j}, V_{ij}, RID_{S_j}) \mid 1 \le j \le n\}$ in mobile device of MU_i.

Step 7. $RC \Longrightarrow MU_i$: A mobile device contains $\{TID_i, (IDS_j, V_{ij}, RID_{S_j}) \mid 1 \le j \le n\}$.

Step 8. MU_i computes $D_i^1 = H(PW_i \parallel \theta_i) \oplus b$, $D_i^2 = H(ID_i \parallel PW_i \parallel \theta_i \parallel b)$, $V_{ij}' = V_{ij} \oplus k = A_{ij} \oplus H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b))$, $RID_{ij} = TID_i \oplus H(ID_i \parallel V_{ij}')$, and $RID_{S_j}' = RID_{S_j} \oplus H(\theta_i \parallel b)$ for $1 \le j \le n$.

Finally, MU_i stores ϕ_i, D_i^1, D_i^2, V_{ij}'s, RID_{ij}s, and RID_{S_j}'s into her own mobile device, and deletes V_{ij}s, TID_i, and RID_{S_j}s from the mobile device.

3.1.2. Cloud Server Registration

Step 1. CS_j chooses her identity ID_{S_j}.

Step 2. $CS_j \Longrightarrow RC : \{ID_{S_j}\}$.

Step 3. RC provides the master secret key X_j to each CS_j.

Step 4. For all MU_is, the RC saves the credentials $\{TID_i, (ID_i, r_{ij})\}$ (for $1 \le i \le m$) in database of CS_j, and also stores $\{ID_{S_j}, X_j\}$ in the database of CS_j.

Finally, RC saves pair (ID_i, SN_i) in its own database, where SN_i is the serial number of MU_i's mobile device.

3.1.3. Login Phase. When MU_i wants to login to CS_j, the following operations shall be executed:

Step L1. MU_i inputs her identity ID_i, password PW_i, and biometrics B_i' into her own mobile device. MU_i computes $\theta_i = Rep(B_i', \phi_i)$ with ϕ_i through the fuzzy extractor reproduction procedure and generates $b' = D_i^1 \oplus H(PW_i \parallel \theta_i)$ with the stored parameter D_i^1, MU_i.

Step L2. MU_i computes $H(ID_i \parallel PW_i \parallel \theta_i \parallel b')$ and checks whether $D_i^2 = H(ID_i \parallel PW_i \parallel \theta_i \parallel b')$. An equality indicates that MU_i is legal.

Step L3. MU_i calculates $RPWB_i = H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b'))$. MU_i also generates $A_{ij} = V_{ij}' \oplus RPBW_i$ using the device parameter V_{ij}'.

Step L4. MU_i selects an 128 bit random number RN_i, generates the current timestamp TS_i, and then computes:

$C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(IDS_j)$, $H_1 = H(ID_i \parallel C_1 \parallel RN_i \parallel TS_i)$, $TID_i = RID_{ij} \oplus H(ID_i \parallel V_{ij}')$, $RID_{S_j} = RID_{S_j}' \oplus H(\theta_i \parallel b')$, and $TID_i^* = TID_i \oplus H(RID_{S_j} \parallel TS_i)$.

Step L5. $MU_i \longrightarrow CS_j : \{TID_i^*, C_1, H_1, TS_i\}$.

3.1.4. Authentication Phase. In this phase, CS_j and MU_i mutually authenticate each other and establish a session key.

Since this phase is unrelated to our discussions, we omit it.

3.2. Flaws in Roy Et Al.'s Scheme. Recall that the three assumptions listed in Section 3 are also clearly made in Roy et al.'s scheme. However, we observe that this scheme still remains feasible for an attacker to offline guess a user's password. This means that the primary goal of "truly two-factor security" cannot be satisfied. In addition, the scheme cannot provide forward secrecy and sound scalability.

3.2.1. Offline Password Guessing Attack. Noting that Roy et al.'s scheme [19] is originally a three-factor one, here we are only interested in its two-factor version by assuming that the third factor (i.e., the biometric) has been known to $\mathscr{A}$. This is realistic as users' biometrics are constant during their lives, and how to protect user biometric template is still an open issue [55]. We find that this scheme cannot achieve truly two-factor security: it is subject to two types of offline password guessing attack. This in turn indicates that it cannot achieve truly three-factor security.

Type-I Attack. Suppose U_i's biometric B_i and the secret parameters $\{D_i^1, D_i^2, \varphi_i, h(\cdot)\}$ stored in the smart card are somehow obtained by $\mathscr{A}$. At this point, $\mathscr{A}$ can find out U_i's identity and password as follows:

Step 1. Guesses U_i's identity ID_i^* and password PW_i^* from dictionary space $\mathscr{D}_{id}$ and $\mathscr{D}_{pw}$.

Step 2. Computes $\theta_i = Rep(B_i, \varphi_i)$, $b^* = H(PW_i \parallel \theta_i) \oplus D_i^1$, where D_i^1 is extracted from the smart card.

Step 3. Computes $D_i^{2*} = H(ID_i^* \parallel PW_i^* \parallel \theta_i \parallel b^*)$, where θ_i is extracted from the smart card.

Step 4. Checks the validity of (ID_i^*, PW_i^*) by comparing the calculated D_i^{2*} with the extracted D_i^2.

Step 5. Repeats Steps 1~4 until finds the correct pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|\mathscr{D}_{id}| * |\mathscr{D}_{pw}| * 2T_H + T_B)$ [3, 16]. Generally, it is only needed to calculate the biohashing function once; thus T_B can be ignored in practice. According to the running time in [16], $\mathscr{A}$ may complete the above attacking procedure within 17.6 days on a Laptop. This issue arises due to the inherent "usability-security tension": to achieve local password change (i.e., C-2 in [3]) and timely typo detection (i.e., C-9 in [3]); there is an explicit password verifier $D_i^2 = H(ID_i \parallel PW_i \parallel \theta_i \parallel b)$ stored in U_i's smart card, yet this verifier leads to a Type-I offline password attack.

To eliminate this security issue without loss of usability, a promising countermeasure is to adopt the "fuzzy-verifier" technique [20] and store $D_i^2 = h((H(ID_i \parallel PW_i \parallel \theta_i \parallel b)) \bmod n)$ in U_i's smart card, where n specifies/confines the capacity of (ID, PW) pair, $2^4 \le n \le 2^8$. As a result, even if $\mathscr{A}$ gets D_i^2, she can not figure out the correct (ID, PW) from the above attack, because there will be $|\mathscr{D}_{id} * \mathscr{D}_{pw}|/n \approx 2^{32}$ candidate (ID, PW) pairs that make $D_i^{2*} = D_i^2$ in Step 4. To further identify the exactly correct (ID, PW) pair, $\mathscr{A}$ needs to interact with the server, and we can adopt the "honeywords" technique [3, 56] to confine $\mathscr{A}$'s advantage to a very limited value.

Type-II Attack. In the above attack, $\mathscr{A}$ does not need the protocol messages. In this attack, we assume that $\mathscr{A}$ can somehow obtain user's smart card and extract its secret parameters $\{D_i^1, D_i^2, V_{ij}', \varphi_i, h(\cdot)\}$ and also can eavesdrop the login messages $\{TID_i^*, C_1, H_1, TS_i\}$ from the public channel. Now, $\mathscr{A}$ can *offline* guess U_i's password and identity simultaneously as follows:

Step 1. Guesses the value of ID_i^*, PW_i^* from dictionary space $\mathscr{D}_{id}$ and $\mathscr{D}_{pw}$.

Step 2. Computes $\theta_i = Rep(B_i, \varphi_i)$, $b^* = H(PW_i \parallel \theta_i) \oplus D_i^1$, where D_i^1 is extracted from the smart card.

Step 3. Computes $A_{ij}^* = V_{ij}' \oplus H(ID_i^* \parallel H(PW_i^* \parallel \theta_i \parallel b^*))$.

Step 4. Computes $RN_i^* = C_1 \oplus A_{ij}^* \oplus TS_i \oplus H(ID_{S_j})$.

Step 5. Computes $H_1^* = H(ID_i^* \parallel C_1 \parallel RN_i^* \parallel TS_i)$.

Step 6. Checks the correctness of (ID_i^*, PW_i^*) by comparing if the computed H_1^* equals the intercepted H_1.

Step 7. Repeats Steps 1~6 of the above procedure until find the correct value of (ID_i^*, PW_i^*).

The time complexity of the above attacking procedure is $\mathcal{O}(|\mathscr{D}_{id}| * |\mathscr{D}_{pw}| * (5T_H + T_B))$, and $\mathscr{A}$ may complete the procedure within 44 days on a Laptop. In comparison, the Type-I attack is more practical.

3.2.2. No Forward Secrecy. A scheme that supports forward secrecy ensures that even after the long-term private key(s) of one or more participants were leaked, previously agreed session keys remain secure [27]. This is important, especially when considering the serious situations of today's clouds like the compromise of cloud servers (e.g., [15]).

If an attacker $\mathscr{M}$ has obtained the server CS_j's long-term key X_j from the breached server S and intercepted the messages $\{TID_i^*, C_1, H_1, TS_i\}$ that are exchanged between U_i and S's authentication process from the public channel. $\mathscr{M}$ is able to figure out the session key using the following method:

Step 1. $\mathscr{M}$ computes $RID_{S_j} = H(ID_{S_j} \parallel X_j)$ and $TID_i = TID_i^* \oplus H(RID_{S_j} \parallel TS_i)$

Step 2. $\mathscr{M}$ computes $A_{ji} = H(H(ID_i \oplus r_{ij}) \parallel X_j)$, $RN_j = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus A_{ji}$, $M_1 = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji}$;

Step 3. $\mathscr{M}$ calculates the session key $sk = H(ID_i \parallel ID_{S_j} \parallel A_{ji} \parallel M_1 \parallel RN_j \parallel TS_i \parallel TS_j)$.

With the session key sk computed, the entire session will be no secret to $\mathcal{M}$.

3.2.3. Poor Scalability. In Roy et al.'s scheme, the user side stores all the cloud servers (i.e., CS_j, $1 \le j \le n$) related information: $RID_{ij} = TID_i \oplus H(ID_i \parallel V'_{ij})$ and $RID'_{S_j} = RID_{S_j} \oplus H(\theta_i \parallel b)$ for $1 \le j \le n$. This means that when a new server arrives, all the users have to reregister with the registration center RC. This shows poor scalability. Similarly, the server side stores all the users (i.e., U_i, $1 \le i \le m$) related information: the credentials $\{TID_i, (ID_i, r_{ij})\}$ (for $1 \le i \le m$) in database of CS_j. This means that when a new user arrives, all the servers have to reregister with the registration center RC.

4. Revisiting Amin Et Al.'s Scheme

In 2018, Amin et al. [21] pointed out that previous schemes (e.g., [35–37]) are vulnerable to various security loopholes and developed a new scheme for distributed cloud computing environments. However, here we show that this scheme still has several serious defects being overlooked (i.e., offline password guessing attack, no forward secrecy, and no user untraceability).

4.1. Review Amin Et Al.'s Scheme

4.1.1. Cloud Server Registration Phase. The registration phase is partitioned into two parts: cloud server registration and user registration. When the cloud server S_m registers, S_m chooses her identity SID_m, a random nonce d, and sends $\langle SID_m, d\rangle$ to CS. Upon getting S_m's registration request, the control server CS computes $PSID_m = h\langle SID_m \parallel d\rangle$, $BS_m = h(PSID_m \parallel y)$, and sends $\langle BS_m\rangle$ to S_m over a secure channel. Finally, S_m keeps the secret data $\langle BS_m, d\rangle$ in a secure place (e.g., a usb-key).

4.1.2. User Registration Phase. When the user U_i registers, she first prefers an identity ID_i, password PW_i, and 2 random nonces $\langle b_1, b_2\rangle$. Then, she computes $A_i = h(PW_i \parallel b_1)$, $PID_i = h(ID_i \parallel b_2)$, $bb_i = b_2 \oplus A_i$, and sends $\langle A_i, PID_i\rangle$ to the CS securely. Once getting $\langle A_i, PID_i\rangle$, CS conducts the following steps:

$$\begin{aligned} C_i &= h\left(A_i \parallel PID_i\right) \\ D_i &= h\left(PID_i \parallel x\right) \\ E_i &= D_i \oplus A_i. \end{aligned} \tag{1}$$

Finally, CS stores the parameters $\{C_i, E_i, h(\cdot)\}$ in the smartcard and delivers the smartcard for U_i through a private communication channel. Then, U_i stores $\langle DP, bb_i\rangle$ in the card, where $DP = h(ID_i \parallel PW_i) \oplus b_1$. Finally, the smartcard stores $\langle C_i, E_i, bb_i, DP, h(\cdot)\rangle$.

4.1.3. Login Phase. U_i first inputs the smartcard into a card reader CR and keys ID_i^* and PW_i^* on the reader. Then, CR calculates

$$\begin{aligned} b_1^* &= DP \oplus h\left(ID_i^* \parallel PW_i^*\right) \\ A_i^* &= h\left(PW_i^* \parallel b_1^*\right) \\ b_2^* &= bb_i \oplus A_i^* \\ PID_i^* &= h\left(ID_i^* \parallel b_2^*\right) \\ C_i^* &= h\left(A_i^* \parallel PID_i^*\right). \end{aligned} \tag{2}$$

Then, the card reader CR checks whether $(C_i^* \stackrel{?}{=} C_i)$. If $(C_i^* == C_i)$, it means that $(ID_i^* = ID_i)$ and $(PW_i^* = PW_i)$. Then, CR produces a 128 bit random nonce N_i and computes

$$\begin{aligned} D_i &= E_i \oplus A_i \\ G_i &= h\left(PID_i \parallel SID_m \parallel N_i \parallel TS_i \parallel D_i\right) \\ F_i &= D_i \oplus N_i \\ Z_i &= SID_m \oplus h\left(D_i \parallel N_i\right) \end{aligned} \tag{3}$$

where SID_m is the identity of the cloud server from which by U_i wants to login. Then, CR sends the login request $\langle G_i, F_i, Z_i, PID_i, TS_i\rangle$ to S_m publicly.

4.1.4. Authentication Phase. This phase is to achieve mutual authentication and key agreement among U_i, S_m, and CS.

Step 1. S_m first verifies whether $(TS_m - TS_i < \triangle T)$, where TS_m is the cloud server's current timestamp and $\triangle T$ is the allowed transmission delay time. If the equality is not true, S_m rejects the connection; otherwise, S_m generates a 128 bit random nonce N_m and computes

$$\begin{aligned} J_i &= BS_m \oplus N_m \\ K_i &= h\left(N_m \parallel BS_m \parallel G_i \parallel TS_j\right). \end{aligned} \tag{4}$$

Then, S_m sends $\langle J_i, K_i, PSID_m, G_i, F_i, Z_i, PID_i, TS_i, TS_m\rangle$ to the CS over a public channel.

Step 2. On getting login messages, CS first checks whether $(TS_{cs} - TS_m < \triangle T^*)$, where TS_{cs} is the cloud server's current timestamp and $\triangle T^*$ the expected delay time for transmission. If the equality is not true, S_m rejects the connection; otherwise, CS computes

$$\begin{aligned} D_i &= h\left(PID_i \parallel x\right) \\ N_i^* &= F_i \oplus D_i \\ SID_m^* &= Z_i \oplus h\left(D_i \parallel N_i^*\right) \\ G_i^* &= h\left(PID_i \parallel SID_m^* \parallel N_i^* \parallel D_i \parallel TS_i\right). \end{aligned} \tag{5}$$

Then, CS checks whether $(G_i^* \stackrel{?}{=} G_i)$. If $(G_i^* == G_i)$, CS deems U_i as legal; otherwise, rejects the connection. Then, CS computes

$$\begin{aligned} BS_m^* &= h\left(PSID_m \parallel y\right) \\ N_m^* &= BS_m^* \oplus J_i \\ K_i^* &= h\left(BS_m^* \parallel N_m^* \parallel G_i \parallel TS_m\right). \end{aligned} \tag{6}$$

Once again, CS checks $(K_i^* \stackrel{?}{=} K_i)$. If $(K_i^* == K_i)$, CS deems S_m as legal; otherwise, it rejects the connection. Then, CS selects a 128 bit random nonce N_{cs} and computes

$$\begin{aligned} P_{cs} &= N_m \oplus N_{cs} \oplus h(N_i \parallel D_i) \\ R_{cs} &= N_i \oplus N_{cs} \oplus h(BS_m^* \parallel N_m^*) \\ SK_{cs} &= h(N_i \oplus N_m \oplus N_{cs}) \\ Q_{cs} &= h((N_m \oplus N_{cs}) \parallel SK_{cs}) \\ V_{cs} &= h((N_i \parallel N_{cs}) \parallel SK_{cs}) \end{aligned} \tag{7}$$

where SK_{cs} is the agreed session key. Then, CS sends $\langle P_{cs}, R_{cs}, Q_{cs}, V_{cs}\rangle$ to S_m.

Step 3. On getting the reply from CS, S_m calculates

$$\begin{aligned} W_m &= h(BS_m \parallel N_m) \\ N_i \oplus N_{cs} &= R_{cs} \oplus W_m \\ SK_m &= h(N_i \oplus N_{cs} \oplus N_m) \\ V_{cs}^* &= h((N_i \oplus N_{cs}) \parallel SK_m). \end{aligned} \tag{8}$$

Then, S_m verifies $V_{cs}^* \stackrel{?}{=} V_{cs}$ or not and if $V_{cs}^* \neq V_{cs}$, rejects the connection, and then sends $\langle P_{cs}, Q_{cs}\rangle$ to U_i publicly.

Step 4. On getting S_m, U_i calculates

$$\begin{aligned} L_i &= h(N_i \parallel D_i) \\ N_m \oplus N_{cs} &= P_{cs} \oplus L_i \\ SK_i &= h(N_m \oplus N_{cs} \oplus N_i) \\ Q_{cs}^* &= h((N_m \oplus N_{cs}) \parallel SK_i). \end{aligned} \tag{9}$$

Then, U_i verifies $(Q_{cs}^* \stackrel{?}{=} Q_{cs})$ and if $(Q_{cs}^* == Q_{cs})$, it demonstrates the authenticity of S_m and CS. Finally, mutual authentication are attained among U_i, S_m and CS. Now, U_i and S_m share the same session key $SK_m = SK_i$, and they can exchange sensitive information subsequently using $SK_m = SK_i$.

4.1.5. Password Change Phase. This phase is performed locally: not interacting with the server. When U_i wants to update her password, she provides ID_i and PW_i after inputting the smartcard. Then, CR executes the operations:

$$\begin{aligned} b_1^* &= DP \oplus h(ID_i^* \parallel PW_i^*) \\ A_i^* &= h(PW_i^* \parallel b_1^*) \\ b_2^* &= bb_i \oplus A_i^* \\ PID_i^* &= h(ID_i^* \parallel b_2^*) \\ C_i^* &= h(A_i^* \parallel PID_i^*). \end{aligned} \tag{10}$$

The smartcard verifies $(C_i^* \stackrel{?}{=} C_i)$. If $(C_i^* == C_i)$, user U_i is asked to enter a new password PW_i^{new} and calculates

$$\begin{aligned} A_i^{new} &= h(PW_i^{new} \parallel b_1) \\ C_i^{new} &= h(A_i^{new} \parallel PID_i^*) \\ D_i &= E_i \oplus A_i = h(PID_i^* \parallel x), \\ bb_i^* &= b_2^* \oplus A_i^{new} \\ E_i^{new} &= D_i \oplus A_i^{new} \\ DP^{new} &= h(ID_i \parallel PW_i^{new}) \oplus b_1^*. \end{aligned} \tag{11}$$

Finally, CR substitutes $\langle C_i, E_i, bb_i, DP^{new}\rangle$ with $\langle C_i^{new}, E_i^{new}, bb_i^*, DP^{new}\rangle$, respectively, in the card.

4.2. Cryptanalysis of Amin Et Al.'s Scheme. Recall that the three assumptions listed in Section 2.2.1 are also clearly made in Amin et al.'s scheme [21]. However, we observe that this scheme still remains feasible for an attacker to offline guess a user's password. This means that the primary goal of "truly two-factor security" cannot be satisfying. In addition, the scheme cannot provide forward secrecy and user untraceability.

4.2.1. Offline Password Guessing Attack. We find Amin et al.'s scheme cannot achieve truly two-factor security: it is subject to two types of offline password guessing attack when the smart card factor is compromised.

Type-I Attack. Suppose the secret parameters $\{C_i, E_i, bb_i, DP, h(\cdot)\}$ stored in U_i's smart card are somehow obtained by $\mathscr{A}$. At this point, $\mathscr{A}$ can find out U_i's identity and password as follows:

Step 1. Guesses U_i's identity ID_i^* and password PW_i^* from dictionary space $\mathscr{D}_{id}$ and $\mathscr{D}_{pw}$.

Step 2. Computes $b_1^* = DP \oplus h(ID_i^* \parallel PW_i^*)$, where DP is extracted from the smart card.

Step 3. Computes $A_i^* = h(PW_i^* \parallel b_1^*)$;

Step 4. Computes $b_2^* = bb_i \oplus A_i^*$, where bb_i is extracted from the smart card.

Step 5. Computes $PID_i^* = h(ID_i^* \parallel b_2^*)$;

Step 6. Computes $C_i^* = h(A_i^* \parallel PID_i^*)$;

Step 7. Checks the validity of (ID_i^*, PW_i^*) by comparing the computed C_i^* with C_i which is extracted from card.

Step 8. Repeats Steps 1~7 until determine the right pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|\mathscr{D}_{id}| * |\mathscr{D}_{pw}| * 4T_H)$ [3, 16]. According to the running time in [16], $\mathscr{A}$ may complete the above attacking procedure within 35.2 days on a Laptop. This issue arises due to the inherent "usability-security tension": to achieve local password change (i.e., C-2 in [3]) and timely typo detection (i.e., C-9 in [3]); there is an

explicit password verifier $C_i = h(A_i \parallel PID_i) = h(h(PW_i \parallel b_1) \parallel h(ID_i \parallel b_2))$ stored in U_i's smart card, yet this verifier leads to a Type-I offline password attack.

To tackle this security issue while not losing usability, a viable countermeasure is to adopt the "fuzzy-verifier" technique [20] and keep $C_i = h(h(PW_i \parallel b_1) \parallel h(ID_i \parallel b_2) \bmod n)$ in U_i's smart card, where n confines the capacity of (ID, PW) pair, $2^4 \leq n \leq 2^8$. As a result, even if $\mathscr{A}$ obtains D_i^2, she can not identify the correct (ID, PW) from the above attack, because there will be $|\mathscr{D}_{id} * \mathscr{D}_{pw}|/n \approx 2^{32}$ candidate (ID, PW) pairs that make $D_i^{2*} = D_i^2$ in Step 4. To further identify the exactly correct (ID, PW) pair, $\mathscr{A}$ needs to interact with the server, and we can adopt the "honeywords" technique [3, 56] to confine $\mathscr{A}$'s advantage to a very limited value.

Type-II Attack. In the above attack, $\mathscr{A}$ does not need the protocol messages. In this attack, we assume that $\mathscr{A}$ can somehow obtain user's smart card and extract its secret parameters $\{C_i, E_i, bb_i, DP, h(\cdot)\}$ and can also eavesdrop the login messages $\{G_i, F_i, Z_i\}$ from the public channel. Now, $\mathscr{A}$ can *offline* guess U_i's password and identity simultaneously as follows:

Step 1. Guesses the value of ID_i^*, PW_i^* from dictionary space $\mathscr{D}_{id}$ and $\mathscr{D}_{pw}$.

Step 2. Computes $b_1^* = DP \oplus h(ID_i^* \parallel PW_i^*)$, where DP is extracted from the smart card.

Step 3. Computes $A_i^* = h(PW_i^* \parallel b_1^*)$;

Step 4. Computes $D_i^* = E_i \oplus A_i^*$.

Step 5. Computes $N_i^* = F_i \oplus D_i^*$;

Step 6. Computes $Z_i^* = SID_m \oplus h(D_i^* \oplus N_i^*)$;

Step 7. Checks the correctness of (ID_i^*, PW_i^*) by comparing if the computed Z_1^* equals the intercepted Z_1.

Step 8. Repeats Steps 1~7 of the above procedure until the correct value of (ID_i^*, PW_i^*) is found.

The time complexity of the above attacking procedure is $\mathcal{O}(|\mathscr{D}_{id}| * |\mathscr{D}_{pw}| * 3T_H)$, and $\mathscr{A}$ may complete the procedure within 26.4 days on a Laptop. In comparison, the Type-I attack is more practical. Amin et al. noted that user's mobile devices and IoT sensors are generally of "low memory, low power, and battery and network limitations" [21], and thus they prefer to only use lightweight symmetric key techniques to achieve robust security. It is just this motivation that makes their scheme vulnerable to the Type-II offline password guessing attack. And the rationales can be found in [20, 57].

4.2.2. No Forward Secrecy. A scheme that supports forward secrecy ensures that even after the long-term private key(s) of one or more participants was(were) leaked, previously agreed session keys remain secure [27]. This is important, especially when considering the serious situations of today's clouds like the compromise of cloud servers (e.g., [15]).

If an attacker $\mathscr{M}$ has obtained the cloud server S_m's long-term key BS_m from the breached server S_m and intercepted the messages $\{J_i; P_{cs}, R_{cs}, Q_{cs}, V_{cs}\}$ that are exchanged between U_i and S_m's authentication process from the public channel. $\mathscr{M}$ is able to figure out the session key using the following method:

Step 1. $\mathscr{M}$ computes $N_m = J_i \oplus BS_m$;

Step 2. $\mathscr{M}$ computes $W_m = h(BS_m \parallel N_m)$;

Step 3. $\mathscr{M}$ computes $N_i \oplus N_{cs} = R_{cs} \oplus W_m$;

Step 4. $\mathscr{M}$ calculates the session key $SK_m = h(N_i \oplus N_{cs} \oplus N_m)$.

With the session key SK_m computed, the entire session will be no secret to $\mathscr{M}$. This vulnerability is due to a result of violating the "forward secrecy principle" given in [20]: public-key technique is necessary to attain forward secrecy and at least two exponential operations (or point multiplications in ECC) are needed at the server side.

4.2.3. No User Untraceability. As revealed in [22], in the context of user authentication, there are two types of user anonymity: (1) the basic one, i.e., user identity-protection which requires that the attacker is unable to get the target user's real identity from eavesdropping the protocol messages; (2) the advanced one, i.e., user untraceability which guarantees that the attacker is unable to neither figure out the target user's identity nor determine whether two conversations are come from the same (unknown) user from eavesdropping the protocol messages. Therefore, most schemes (e.g., [4, 35, 39, 58–62]) that aim to preserve user anonymity attempt to fulfil the latter advanced notion of user anonymity. However, in Amin et al.'s scheme [21], the pseudoidentity PID_i is sent in every login request. Thus, the attacker can track all the login activities through this static pesudoidentity PID_i.

5. Revisiting Wei Et Al.'s Scheme

In 2018, Wei et al. [23] observed that most of previous schemes (e.g., [36–40]) only provided some heuristic security arguments and little attention has been to the formal treatment of protocol security. Accordingly, they employed an authenticated encryption scheme to construct a new two-factor scheme that can provide provable security in the random oracle model. Though this scheme is armed with a formal proof, we will show that this scheme still has several serious defects being overlooked: (1) it cannot achieve the primary goal of "truly two-factor security"; (2) it provides no forward secrecy; (3) it ensures no user untraceability.

5.1. Review Wei Et Al.'s Scheme. There are three phases in et al.'s scheme: registration, authentication and key exchange, and password update.

5.1.1. Registration Phase. When the user U_i registers, she first prefers an identity ID_i, password PW_i, and a random nonces b; then U_i calculates $PW_i^* = h(PW_i \parallel b)$ and sends $\{ID_i, PW_i^*\}$ to the gateway node GW via a secure channel. On receiving $\{ID_i, PW_i^*\}$, GW computes $V_i = h(ID_i \parallel K)$ and $N_i = V_i \oplus h(ID_i \parallel PW_i^*)$. GW also selects U_i's pseudoidentity DID_i and keeps the item (DID_i, ID_i) in its database. Finally, GW produces a smart card with security parameters $(DID_i, N_i, h(\cdot))$ and sends to U_i the card via a trusted channel. Then, U_i adds the random number b into the card.

5.1.2. Authentication and Key Exchange Phase. U_i aims to access the service from the cloud server S_j, U_i executes the authentication and key exchange phase with GW and S_j as follows.

(1) U_i inserts her card into the card reader, keys her identity ID_i, and password PW_i. The card computes $PW_i^* = h(PW_i \parallel b)$ and gets $V_i^* = N_i \oplus h(ID_i \parallel PW_i^*)$. Then, the card calculates a key $k_1 = h(V_i^* \parallel T_1)$, where T_1 is the current timestamp in U_i's system. The card also selects a random nonce $R_1 \in \{0,1\}^l$ and encrypts R_1 using an authenticated encryption scheme to obtain $C_1 = E_{k_1}^{T_1, DID_i}(R_1)$. Finally, the card transmits the message (DID_i, S_j, C_1, T_1) to GW.

(2) Upon receiving (DID_i, S_j, C_1, T_1) at time T_1^*, GW verifies whether $T_1^* - T_1 \leq \Delta T$ or not, where ΔT stands for the largest time interval allowed for the transmission delay. If the check passes, GW retrieves U_i's real identity ID_i in the database using DID_i and calculates $k_1 = h(h(ID_i \parallel K) \parallel T_1)$. GW then decrypts C_1 and recovers $R_1 = D_{k_1}^{T_1, DID_i}(C_1)$. If the decryption fails, the connection is rejected; otherwise, U_i is verified by GW. GW calculates a key $k_2 = h(ID \parallel h(K \parallel S_j) \parallel T_2)$, where T_2 is the current timestamp at GW. Then, GW encrypts R_1 using an authenticated encryption scheme to obtain $C_2 = E_{k_2}^{T_2, ID}\{R_1\}$. Finally, GW transmits (ID, C_2, T_2) to the server S_j.

(3) On getting (ID, C_2, T_2) at time T_2^*, the server S_j verifies if $T_2^* - T_2 \leq \Delta T$. If it is true, S_j calculates $k_2 = h(ID \parallel x_j \parallel T_2)$ and decrypts C_2 and calculates $R_1 = D_{k_2}^{T_2, ID}\{C_2\}$. If the decryption fails, the connection is rejected; otherwise, S_j computes $k_3 = h(ID \parallel R_1 \parallel T_3)$, where T_3 is the current timestamp at S_j. Then, S_j selects a random number $R_2 \in \{0,1\}^l$ and encrypts R_2 using k_3 to get $C_3 = E_{k_3}^{T_3, ID}\{R_2\}$. S_j transmits (C_3, T_3) to U_i. Finally, S_j calculates the session key $sk = h(ID \parallel T_3 \parallel R_1 \parallel R_2)$ and accepts U_i.

(4) On getting (C_3, T_3) at time T_3^*, U_i checks if $T_3^* - T_3 \leq \Delta T$. If it holds, GW calculates $k_3 = h(ID \parallel R_1 \parallel T_3)$ and decrypts $R_2 = D_{k_3}^{T_3, ID}\{C_4\}$. If the decryption does not fail, U_i deems the session as valid and computes $sk = h(ID \parallel T_3 \parallel R_1 \parallel R_2)$.

Finally, U_i and S_j share the common session key sk.

5.1.3. Password Updating Phase. This phase is executed when U_i aims to update her password PW_i with a new one. Because this phase has little relevance with the following cryptanalysis, we omit it.

5.2. Cryptanalysis of Amin Et Al.'s Scheme. Recall that the three assumptions listed in Section 2.2.1 are also clearly made in Wei et al.'s scheme [23]. Though this scheme enjoys many desirable attributes such as rigorous security model and the introduction of an authenticated encryption scheme (i.e., Hoang et al.'s AEZ [63]), we now show its defects.

5.2.1. Offline Password Guessing Attack. We find Amin *et al.*'s scheme cannot achieve truly two-factor security: it is subject to Type-II offline password guessing attack as illustrated in Section 4.2.1.

Type-II Attack. In this attack, we assume that $\mathcal{A}$ can somehow obtain user's smart card and extract its secret parameters $\{DID_i, N_i, b, h(\cdot)\}$ and also can eavesdrop the login messages $\{DID_i, S_j, C_1, T_1\}$ from the public channel. Now, $\mathcal{A}$ can *offline* guess U_i's password and identity simultaneously as follows:

Step 1. Guesses the value of ID_i', PW_i' from dictionary space $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$.

Step 2. Computes $PW_i^{*\prime} = h(PW_i' \parallel b)$, where b is extracted from the smart card.

Step 3. Computes $V_i' = N_i \oplus h(ID_i' \parallel PW_i^{*\prime})$, where N_i is extracted from the smart card.

Step 4. Computes $k_1' = h(V_i' \parallel T_1)$;

Step 5. Decrypts C_1 with k_1';

Step 6. If the above decryption succeeds, it indicates that ID_i' and PW_i' are correct and terminate.

Step 7. Repeats Steps 1~5 of the above procedure until find the correct value of (ID_i, PW_i).

The time complexity of the above attacking procedure is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * 3T_H)$, and $\mathcal{A}$ may finish the attack within 26.4 days on a common PC. In comparison, the Type-I attack is more realistic. Wei et al. noted that user's mobile devices and IoT sensors are generally resource-constrained and made an attempt to design a secure and efficient scheme "without using computation-expensive public key operations" [23]. It is just this motivation that makes their scheme vulnerable to the Type-II offline password guessing attack. And the rationales are referred to [20, 57].

5.2.2. No Forward Secrecy. As with Amin et al.'s scheme [21], Wei et al.'s scheme also only employs symmetric key techniques and thus violates the "forward secrecy principle" proposed in [20], and see more in Section 4.2.2. We now show the details.

Table 2: Summary of our cryptanalysis results and the underlying vulnerabilities.

Scheme	Weaknesses	Vulnerabilities
Wu-Xu [17]	Offline password guessing attack	When using Elgamal-like encryption (e.g., chaotic-based), at least two exponentiations are needed at the user side [18]
	Replay attack	Their scheme employs random numbers but has only two protocol flows
Roy et al. [19]	Offline password guessing attack	No public-key technique used [20]
	No forward secrecy	No public-key technique used [20]
	Poor scalability	Poor design
Amin et al. [21]	Offline password guessing attack	No public-key technique used [20]
	No forward secrecy	No public-key technique used [20]
	No user un-traceability	No public-key technique used [22]
Wei et al. [23]	Offline password guessing attack	No public-key technique used [20]
	No forward secrecy	No public-key technique used [20]
	No user un-traceability	No public-key technique used [22]

If an attacker $\mathcal{M}$ has obtained the cloud server S_j's long-term key x_j from the breached server S_j and intercepted the messages $\{ID, C_2, T_2, T_3, C_3\}$ that are exchanged among U_i, GW, and S_j's authentication process from the public channel, $\mathcal{M}$ is able to figure out the session key using the following method:

Step 1. $\mathcal{M}$ computes $k_2 = h(ID \parallel x_j \parallel T_2)$;

Step 2. $\mathcal{M}$ decrypts C_2 using k_2 to obtain $R_1 = D_{k_2}^{T_2, ID}\{C_2\}$;

Step 3. $\mathcal{M}$ calculates $k_3 = h(ID \parallel R_1 \parallel T_3)$;

Step 4. $\mathcal{M}$ decrypts C_3 using k_3 to obtain $R_2 = D_{k_3}^{T_3, ID}\{C_3\}$;

Step 5. $\mathcal{M}$ calculates the session key $sk = h(ID \parallel T_3 \parallel R_1 \parallel R_2)$.

5.2.3. No User Un-Traceability. As with Amin et al.'s scheme [21], Wei et al.'s scheme also only employs the pseudo-identity technique to preserve user anonymity, and this breaches the "anonymity principle" proposed in [22]: under the non-tamper resistant assumption of smart cards, public-key primitives are necessary for achieving user un-traceability for two-factor user authentication schemes. In Wei *et al.*'s scheme [21], the static pseudo-identity DID_i is sent in every login request. Thus, the attacker can track all the login activities through this static pesudo-identity DID_i.

6. Conclusion

A large amount of efforts have been devoted to the design of smart-card-based password authentication scheme, yet it still remains an open challenge to devise an efficient, secure, and privacy-preserving scheme under the assumption that smart cards can be tampered. Very recently, Wu-Xu, Roy et al., Amin et al., and Wei et al. made four other attempts. However, through careful analysis we show that all of them still have several serious drawbacks being overlooked (see Table 2), and most these drawbacks are due to violation of basic protocols design principles (e.g., [3, 20]). Taking our attacks in mind, we are considering designing an efficient two-factor scheme with provable security.

Data Availability

Data sharing is not applicable to this article as no new data was created or analyzed in this study.

Disclosure

Part of this paper was presented at the 4th International Conference on Cloud Computing and Security (ICCCS 2018) [14].

Conflicts of Interest

The authors declare that no conflicts of interest exist.

Acknowledgments

This research was in part supported by the National Key Research and Development Plan under Grant No. 2016YFB0800600, by the National Natural Science Foundation of China under Grant No. 61802006, by the National Key Research and Development Plan under Grant No. 2017YFB1200700, and by China Postdoctoral Science Foundation under Grant No. 2018M640026.

References

[1] Z. Hao, S. Zhong, and N. Yu, "A time-bound ticket-based mutual authentication scheme for cloud computing," *International Journal of Computers, Communications and Control*, vol. 6, no. 2, pp. 227–235, 2011.

[2] J. Yu, G. Wang, Y. Mu, and W. Gao, "An efficient generic framework for three-factor authentication with provably secure

instantiation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2302–2313, 2014.

[3] 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.

[4] 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.

[5] 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.

[6] R. Amin, S. H. Islam, P. Gope, K. R. Choo, and N. Tapas, "Anonymity preserving and lightweight multi-medical server authentication protocol for telecare medical information system," *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2018.

[7] 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, pp. 1–20, 2019.

[8] 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.

[9] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K. R. Choo, and Y. Park, "Design of secure and lightweight authentication protocol for wearable devices environment," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1310–1322, 2018.

[10] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: A comparative evaluation of two-factor authentication schemes," in *Proceedings of the ACM, ASIACCS '16*, pp. 475–486, 2016.

[11] H. Xiong, J. Tao, and C. Yuan, "Enabling telecare medical information systems with strong authentication and anonymity," *IEEE Access*, vol. 5, pp. 5648–5661, 2017.

[12] D. Wang and P. Wang, "On the usability of two-factor authentication," in *Proceedings of the SecureComm*, pp. 141–150, 2014.

[13] G. Xu, J. Liu, Y. Lu, X. Zeng, Y. Zhang, and X. Li, "A novel efficient MAKA protocol with desynchronization for anonymous roaming service in Global Mobility Networks," *Journal of Network and Computer Applications*, vol. 107, pp. 83–92, 2018.

[14] Y. Shen, D. Wang, and P. Wang, "Revisiting anonymous twofactor authentication schemes for cloud computing," in *Proceedings of the ICCCS*, pp. 134–146, 2018.

[15] All Data Breach Sources, https://breachalarm.com/all-sources, 2019.

[16] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous twofactor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.

[17] F. Wu and L. Xu, "A chaotic map-based authentication and key agreement scheme with user anonymity for cloud computing," in *Proceedings of the ICCCS '17*, pp. 189–200, Springer, 2017.

[18] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: efficient and provably secure twofactor authentication scheme with user anonymity," *Information Sciences*, vol. 321, pp. 162–178, 2015.

[19] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, no. 25, pp. 808–25825, 2017.

[20] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2215–2227, 2014.

[21] 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.

[22] 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.

[23] F. Wei, R. Zhang, and C. Ma, "A provably secure anonymous two-factor authenticated key exchange protocol for cloud computing," *Fundamenta Informaticae*, vol. 157, no. 1, pp. 201–220, 2018.

[24] G. Yang, D. S. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *Journal of Computer and System Sciences*, vol. 74, no. 7, pp. 1160–1172, 2008.

[25] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1767–1775, 2014.

[26] X. Li, J. Niu, J. Liao, and W. Liang, "Cryptanalysis of a dynamic identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 374–382, 2015.

[27] C. Wang and G. Xu, "Cryptanalysis of three passwordbased remote user authentication schemes with nontamper resistant smart card," *Security and Communication Networks*, vol. 2017, Article ID 1619741, 14 pages, 2017.

[28] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[29] S. K. Sood, A. K. Sarje, and K. Singh, "An improvement of xu et al.'s authentication scheme using smart cards," in *Proceedings of the ACM COMPUTE '10*, pp. 1–5, ACM, 2010.

[30] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.

[31] B.-L. Chen, W.-C. Kuo, and L.-C. Wuu, "Robust smart-card-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 27, no. 2, pp. 377–389, 2014.

[32] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the CRYPTO 1999*, vol. 1666 of *LNCS*, pp. 388–397, Springer, 1999.

[33] 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.

[34] 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.

[35] K.-P. Xue, P.-L. Hong, and C.-S. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[36] X. Li, Y. Xiong, J. Ma, and W. Wang, "An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

[37] B. Wang and M. Ma, "A smart card based efficient and secured multi-server authentication scheme," *Wireless Personal Communications*, vol. 68, no. 2, pp. 361–378, 2013.

[38] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.

[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] D.-Z. Sun, J.-X. Li, Z.-Y. Feng, Z.-F. Cao, and G.-Q. Xu, "ON the security and improvement of a two-factor user authentication scheme in wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 895–905, 2013.

[41] D. Dolev and A. C.-C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[42] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383–393, 2014.

[43] 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 protection for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.

[44] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.

[45] D. Wang, P. Wang, H. Debiao, and Y. Tian, "A security analysis of honeywords," in *Proceedings of the Usenix Security*, pp. 1–18, 2019.

[46] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proceedings of the IEEE S&P*, pp. 538–552, 2012.

[47] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proceedings of the ACM CCS*, pp. 1242–1254, 2016.

[48] D. Wang and P. Wang, "Offline dictionary attack on password authentication schemes using smart cards," in *Proceedings of the ISC*, pp. 221–237, 2013.

[49] S. H. Islam, "Design and analysis of an improved smartcard-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 29, no. 11, pp. 1708–1719, 2016.

[50] M. Alsaleh, M. Mannan, and P. C. van Oorschot, "Revisiting defenses against large-scale online password guessing attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 128–141, 2012.

[51] P. A. Grassi, E. M. Newton, R. A. Perlner et al., "NIST 800-63B digital identity guidelines: Authentication and lifecycle management," Tech. Rep., McLean, Va, USA, 2017.

[52] AD. Jaggard and P. Syverson, "Oft target," in *Proceedings of the PET*, pp. 1-2, 2017.

[53] B. Lu, X. Zhang, Z. Ling, Y. Zhang, and Z. Lin, "A measurement study of authentication rate-limiting mechanisms of modern websites," in *Proceedings of the ACSAC*, pp. 89–100, 2018.

[54] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223–230, 2012.

[55] N. Memon, "How biometric authentication poses new challenges to our security and privacy [in the spotlight]," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 194–196, 2017.

[56] D. Wang, H. Cheng, P. Wang, J. Yan, and X. Huang, "A security analysis of honeywords," in *Proceedings of the NDSS*, ISOC, pp. 1–16, 2018.

[57] 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.

[58] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient passwordauthenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.

[59] 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*, 2019.

[60] A. Gupta, M. Tripathi, T. J. Shaikh, and A. Sharma, "A lightweight anonymous user authentication and key establishment scheme for wearable devices," *Computer Networks*, vol. 149, pp. 29–42, 2019.

[61] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. Rodrigues, "Biometrics-Based Privacy-Preserving User Authentication Scheme for Cloud-Based Industrial Internet of Things Deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.

[62] T.-T. Truong, M.-T. Tran, and A.-D. Duong, "Improved chebyshev polynomials-based authentication scheme in client-server environment," *Security and Communication Networks*, vol. 2019, Article ID 4250743, 11 pages, 2019.

[63] V. T. Hoang, T. Krovetz, and P. Rogaway, "Robust authenticated-encryption aez and the problem that it solves," in *Proceedings of the EUROCRYPT*, pp. 15–44, 2015.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 4752612, 11 pages
https://doi.org/10.1155/2019/4752612

Research Article

A Practical Authentication Framework for VANETs

Baosheng Wang, Yi Wang, and Rongmao Chen

School of Computer, National University of Defence Technology, 410073, China

Correspondence should be addressed to Rongmao Chen; chromao@nudt.edu.cn

Received 14 March 2019; Revised 29 April 2019; Accepted 7 May 2019; Published 20 May 2019

Guest Editor: Fagen Li

In vehicular ad hoc networks (VANETs), conditional privacy preserving authentication (CPPA) scheme is widely deployed to solve security and privacy issues. Existing CPPA schemes usually require ideal tamper-proof devices (TPDs) on vehicles which, however, might be infeasible or do not exist in reality due to high security requirements. To address this problem, we propose a practical framework of CPPA scheme that supports more realistic TPDs which are less secure correspondingly. We demonstrate that this framework also manages to achieve nonframeability in addition to other security objectives including nonrepudiation, conditional privacy preserving, and unlinkability. Moreover, performance analysis shows that our framework has better efficiency in authentication. All these features make our framework practical for VANETs.

1. Introduction

As one form of mobile ad hoc network in the domain of vehicles, vehicular ad hoc network (VANET) is a promising solution for improving road safety and driving experience. Generally, a VANET is composed of roadside units (RSUs) and vehicles equipped with electronic components such as wheel rotation sensors, radars, and on-broad units (OBUs). Various sensors on vehicle provide continuous monitoring of driving information, such as speed, direction, and position. OBUs enable vehicles to communicate with not only each other but also RSUs via Dedicated Short Range Communications (DSRC) technique. Thus, there are plenty of potential applications on VANETs which can be categorized into safety-related, such as collision avoidance and automatic driving, and other applications, such as traffic navigation and infotainment.

For the security of VANET and its applications, especially safety-related applications, it is crucial to authenticate transmitted messages and identities of their senders; otherwise any unauthorized vehicle could disseminate bogus messages easily or conduct other malicious behaviours without being caught, which might cause great damages to urban transportation systems and even endanger the lives of drivers and pedestrians. To authenticate itself to other entities, vehicle might have to prove the possession of secret information which is usually saved in tamper-proof device (TPD) on vehicle. In addition to storage of secret data, TPD also provides computation service where secret information is involved. For instance, the simplest way to achieve authentication is using digital signature. Every vehicle is assigned to a public/private key pair, and TPD is responsible for storing private keys and generating signatures. Many authentication schemes [1–8] are designed under the assumption of using ideal TPD that can never be compromised by adversary to securely store secrets and to perform related calculations. However, this assumption might be too strong to be realistic in practice. Specifically, in VANET conditions, TPD might mistake normal shocks of vehicle caused by uneven road surface for malicious tampering and erase all the secrets [9]. Moreover, it is possible for adversary to collect sufficient information about secrets in TPD through side-channel attacks such as electromagnetic radiation [10] and power consumption analysis [11].

To address this problem, we loosen the security requirements on ideal TPD and consider a more realistic TPD for practical use. Comparing to ideal TPD, realistic TPD is less sensitive to vehicle shocks but might be compromised by sophisticated hardware tampering. To cope with such hardware tampering as well as aforementioned side-channel attacks, we assume that realistic TPD offers temporary storage of secrets and erases them regularly before adversary obtains

substantial information about them. In this work, we propose an efficient framework of CPPA scheme based on identity-based cryptography (IBC) that only requires realistic TPD.

Our framework also aims at achieving nonframeability [12]. That is, trusted authority (TA) that serves particular region as certification authority and RSUs cannot forge messages to frame an innocent vehicle. TA in existing works (e.g., [6, 13]) usually holds all the secrets of vehicles, so it is quite simple for unrestricted TA to impersonate any vehicle and forge its signature. In our framework, the key used for authentication is independently generated by vehicle itself and stored in TPD. TA that does not possess the authentication key of vehicle cannot impersonate vehicle and successfully authenticate itself to RSU. Meanwhile, RSU's master key which is used to generate the signatures of messages sent by vehicles in our framework is unknown to TA. Thus, TA also cannot forge the signature of vehicle. Besides, RSU cannot forge it either as the pseudo-identity generation also requires vehicle's authentication key.

We design our framework with an objective of improving the efficiency of mutual authentication between vehicle and RSU. Since the location and identity of RSU are relatively fixed, RSU-to-vehicle (R2V) authentication is rather trivial and can be efficiently achieved by periodically broadcasting signed messages. However, vehicle-to-RSU (V2R) authentication in existing works (e.g., [13, 14]) needs the cooperation of TA. In contrast, V2R authentication in our framework does not require real-time interactions between RSUs and TA. Precisely, TA maintains a dynamic list that contains authentication-related information of vehicles, and every RSU is asked to store a latest copy of this list in the background. This list enables RSUs to complete the anonymous authentication of vehicles by themselves, which reduces the workload of TA and promotes the efficiency of authentication. Generally, the main contributions of our work are as follows.

(i) We propose an efficient IBC-based framework of CPPA scheme to solve security and privacy issues in VANETs. Due to the support of realistic TPD, our framework is a practical authentication solution in reality.

(ii) Our framework has achieved nonframeability. The authentication of vehicle and the generation of a valid signature both require vehicle's self-chosen authentication key, which prevents TA and RSUs from framing an innocent vehicle.

(iii) In our framework, we design a mechanism to improve the efficiency of authentication. The overall workload of authentication is distributed to every RSU. Instead of participating in the process of authentication directly, TA just needs to maintain the latest information list for RSUs.

(iv) We give a specific analysis of our framework in terms of security and performance. We prove that this framework has achieved all the security objectives in Section 3. Theoretical analysis on performance indicates that this framework provides excellent authentication efficiency.

The rest of the paper is organized as follows. Section 2 summarizes the related work on authentication schemes for VANETs. Section 3 introduces the architecture of VANETs and our design goals. Preliminary background of cryptographic primitives is provided in Section 4. In Section 5, we present our framework of CPPA scheme. Sections 6 and 7 give the comprehensive security analysis and performance evaluation of our framework, respectively. Section 8 concludes the paper.

2. Related Work

A number of related studies have been reported on authentication issue in VANETs, and their proposed authentication schemes can be categorized into following four types.

Schemes based on Public Key Infrastructure (PKI) [9, 15]. PKI issues a bunch of public/private key pairs and public key certificates to vehicles. Before sending a message, vehicle has to attach a digital signature and a certificate to it, which might increase the communication overhead significantly. To achieve identity privacy and conditional anonymity, anonymous public keys are required for PKI and vehicles. The management of certificates including revocation could be a heavy burden to PKI.

Schemes based on symmetric cryptosystem [12, 16, 17]. Message authentication code (MAC) can be adopted to authenticate message and the verification of the message can be completed in extremely short time. However, the process of message authentication might need the aid of RSUs, and vehicle cannot authenticate received message independently. TESLA [18] is an efficient broadcast authentication protocol based on MAC and loose time synchronization between network nodes. Based on TESLA and the prediction of vehicle direction, it is possible to achieve instant verification of beacon messages sent by vehicles. Unfortunately, this protocol allows adversary to trace the trajectory of vehicle.

Schemes based on group signature [1, 4, 19–22]: group signature naturally provides privacy to group members because every member signs message on behalf the group. The group manager owns the master key of group and is able to learn the real identities of group members, which satisfies the requirement of conditional privacy preservation. However, the verification of group signature usually costs more time than that of traditional signature. Also, revoking compromised group members properly is still a problem.

Schemes based on IBC [2, 6, 7, 13, 14, 23–25]. In identity-based signature (IBS) scheme, the identity of vehicle could be used as the public key, and the corresponding private key is generated by the private key generator (PKG) using master key. Comparing to PKI, it avoids the management of certificates. To achieve conditional privacy, vehicles communicate with other entities using pseudo-identities that are retrievable to authorities. Unfortunately, due to bilinear pairing operations, the time efficiency of IBS schemes is relatively lower than other traditional signature schemes.

To improve the performance, batch verification is adopted to verify multiple signatures at the same time. Moreover, efficient one-time IBS [6, 13], identity-based online/offline signature (IBOOS) [7], and IBS without bilinear pairing [6] also are used in authentication schemes.

3. Background

3.1. Network Architecture. A VANET commonly consists of vehicles, RSUs, and TA, as shown in Figure 1.

TA plays the role of administrator in VANET and manages the authentication of network nodes including vehicles and RSUs. To join the VANET, all the nodes must register themselves at TA in advance. Due to the mobility of vehicles, we consider a frequently changing group of vehicles that requires TA to provide real-time registration service via secure network infrastructure. In contrast, the locations and total number of RSUs usually stay unchanged for a relatively long period of time. The registration of RSUs can be finished during initialization phase. Also, TA maintains a list of registered vehicles and has responsibility for revealing real identities of misbehaving vehicles and revoking licenses of these vehicles in time.

RSUs as roadside infrastructure are scattered all over the region of TA. Communication between RSU and TA relies on wired channel while RSU communicates with vehicles via wireless channel using DSRC protocol. RSUs forward messages not only between TA and vehicles but also from one vehicle to another. A RSU and vehicles enrolled by it form a subgroup of VANET. Vehicles that newly enter the transmission range of RSU have to be authenticated by RSU.

Every vehicle is equipped with OBU to communicate with other entities in VANET and support DSRC protocol. Realistic TPD is also embedded in vehicle. It provides temporary storage of secret information and related computation service, which is more feasible than ideal TPD that never discloses any secrets. Therefore, secrets stored in TPD needs to be updated regularly with the assistance of TA.

3.2. Design Goals. As a framework of CPPA scheme, our framework should satisfy basic requirements: authentication, nonrepudiation, identity privacy preserving, and conditional traceability.

(i) *Authentication*: there are two kinds of authentication: message and entity authentication. Message authentication is confirming that received messages are generated by valid vehicles and unmodified during transmission. Entity authentication, also called mutual authentication, requires that two entities into a session are able to identify each other.

(ii) *Nonrepudiation*: this property refers to a situation where a receiver is able to prove to a third party that sender cannot deny its responsibility for generating messages. It prevents adversary from forging messages in other identities.

(iii) *Identity privacy preserving*: vehicles on the roads are required to frequently broadcast messages including position, speed, direction, and driving status. Identity privacy preservation means that nobody could discover the binding between messages and real identities of vehicles.

(iv) *Conditional traceability*: in certain circumstances (e.g., traffic accidents), the real identities of vehicles should be retrievable. Conditional traceability enables TA only to recover the real identities of vehicles from saved messages.

Considering the particular scenario of VANET, we also attempt to achieve other meaningful properties at the same time.

(i) *Nonframeability*: this property requires no entities in VANETs including TA and RSUs could frame an innocent vehicle or accuse an honest vehicle for having misbehaved. To achieve this security goal, we assume that TA does not collude with RSUs.

(ii) *Ideal TPD freeness*: under the premise of ensuring system security, this property proposed by Zhang et al. [13] permits the usage of realistic TPD or one with sufficient security level embedded in vehicle, instead of ideal one which can never be compromised by adversary.

(iii) *Unlinkability*: let m_1 and m_2 be two messages sent by one vehicle; this property means that one cannot determine whether m_1 and m_2 originate from the same vehicle or not. Unlinkability prevents adversary from tracking vehicles and profiling drivers.

(iv) *Message confidentiality*: in particular applications, messages should be transmitted to receivers in encrypted form and cannot be decoded by unauthorized entities.

(v) *Attack resistance*: this property requires that proposed framework can withstand common attacks, such as replay attack, impersonation attack, modification attack, and side-channel attack.

4. Preliminaries

4.1. Cryptographic Schemes. A symmetric encryption scheme consists of three algorithms which are described as follows.

(i) KeyGen(1^n): this algorithm takes as input security parameter 1^n and outputs key $K \in \mathcal{K}$, where $\mathcal{K}$ is the key space.

(ii) Enc(K, m): this algorithm takes as input key K and message m and outputs ciphertext c.

(iii) Dec(K, c): this algorithm takes as input key K and ciphertext c and outputs message m.

An identity-based signature (IBS) scheme is composed of four algorithms which are described as follows.

(i) Setup(1^n): this algorithm takes as input security parameter 1^n and generates the public parameters PP and master key $msk \in \mathcal{MK}$ for private key generator (PKG). Note that msk is kept secret.

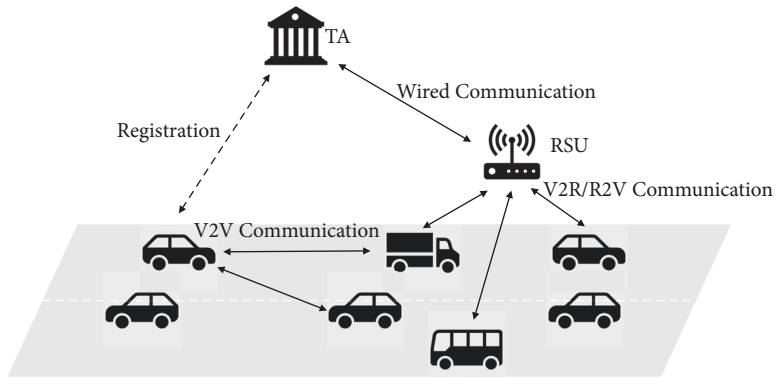


Figure 1: The network architecture of VANET.

(ii) Extract(msk, ID): this algorithm takes as input master key msk and an identity and outputs a private key $sk_{ID} \in \mathcal{SK}$.

(iii) Sign(sk_{ID}, m): this algorithm takes as input private key sk_{ID} and message m and generates a signature $Sign$ of message m.

(iv) Verify($PP, ID, m, Sign$): this algorithm outputs "accept" if $Sign$ is valid signature of message m and outputs "reject" otherwise.

An identity-based online/offline signature (IBOOS) scheme is an IBS scheme where the process of generating signature can be divided into offline and online phases:

(i) $\text{Sign}_{\text{off}}(PP)$: based on public parameters PP, this algorithm generates an offline signature $\overline{Sign}$.

(ii) $\text{Sign}_{\text{on}}(sk_{ID}, \overline{Sign}, m)$: based on private key sk_{ID}, offline signature $\overline{Sign}$, and message m, this algorithm generates a signature $Sign$ of message m.

An one-time identity-based signature (OT-IBS) scheme is an IBS scheme with one-time private key sk_{ID}. Similar to signing key in one-time signature scheme, every private key in OT-IBS can be used only once.

4.2. Cryptographic Hardness Assumption. Computational Diffie-Hellman (CDH) assumption: let $\mathbb{G}$ be a group with prime order p and $g \in \mathbb{G}$ is a random generator of $\mathbb{G}$, and $\mathcal{A}$ is a probabilistic polynomial-time (PPT) algorithm that takes as input a tuple (g, g^a, g^b) and outputs g^c. We define the CDH-advantage of $\mathcal{A}$ to be $Adv_{\mathcal{A}}^{CDH}(n) = \Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}]$. The CDH assumption is that there is no PPT algorithm $\mathcal{A}$ that can compute g^{ab} with nonnegligible CDH-advantage.

5. Proposed Framework of CPPA Scheme

5.1. Overview. In initialization phase of our framework, TA generates parameters for the whole system. RSUs and vehicles are allowed to join VANET after registration. For vehicle that drives into a new RSU region, it also needs to conduct mutual authentication with RSU. To conceal the real identity of vehicle from RSU, V2R authentication needs the assistance of a list maintained by TA that consists of authentication-related information of vehicles. If this authentication succeeds, vehicle would receive the master key of RSU and be able to sign messages in pseudo-identities. Only TA can recover the real identity of vehicle from its pseudo-identities. There also is an efficient and secure mechanism of updating secrets (i.e., authentication key of vehicle and master key of RSU) in TPD before adversary has collected sufficient information via side-channel attacks. Notations used in our framework are defined as follows.

(i) $\mathcal{IBS}$ = (Setup, Extract, Sign, Verify): an IBS scheme that supports batch verification of multiple signatures.

(ii) $\mathcal{SE}$ = (KeyGen, Enc, Dec): a symmetric encryption scheme with message space $\{0,1\}^*$ and key space $\mathcal{K}$.

(iii) $\mathbb{G}_1, \mathbb{G}_2$: two cyclic groups with prime order q.

(iv) g_1, g_2: two generators of $\mathbb{G}_1, \mathbb{G}_2$.

(v) H_1, H_2, H_3, H_4: four hash functions $H_1 : \{0,1\}^* \longrightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \longrightarrow \mathcal{K}$, $H_3 : \{0,1\}^* \longrightarrow \{0,1\}^\ell$, and $H_4 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$.

(vi) ID_{U_i}/ID_{V_j}: real identity of RSU U_i or vehicle V_j.

(vii) PID_{V_j}: pseudo-identity of vehicle V_j.

5.2. System Initialization. In initialization phase, TA generates parameters for the whole system and all the RSUs and vehicles have to register themselves to TA before joining the VANET. Precisely, the system is initialized as follows.

TA Setup: TA runs algorithm Setup to generate public parameters PP_s and system master key msk_s. TA also generates two cyclic groups $\mathbb{G}_1, \mathbb{G}_2$with prime order q and picks generators $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$. Then it picks $x, y, z \in \mathbb{Z}_q^*$ and computes $X = g_1^x, Y = g_1^y, Z = g_1^z$ which are used to generate pseudo-identities for vehicle. Hash functions $H_1 : \{0,1\}^* \longrightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \longrightarrow \mathcal{K}$, $H_3 : \{0,1\}^* \longrightarrow \{0,1\}^\ell$, and $H_4 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ are chosen by TA. The system public parameters are $PP = (PP_s, q, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, X, Y, Z, H_1, H_2, H_3, H_4)$. TA also maintains a private list $\mathcal{L}$ to record authentication

information of registered vehicles as well as list $\mathscr{L}_{pub}$ that is only accessible to registered RSUs. The details of these lists are described later.

RSU Setup: since TA is the only party that owns msk_s in current system, RSU U_i with identity ID_{U_i} obtains its private key $sk_{U_i} \longleftarrow \text{Extract}(msk_s, ID_{U_i})$ from TA. Besides, each RSU U_i has to generate their own public parameters PP_{U_i} and master key msk_{U_i} by running algorithm Setup. For the sake of system security, we require RSU to update its public parameters and master key regularly and share its latest public parameters among all the registered RSUs.

Vehicle Setup: vehicle should register itself to local TA via secure network infrastructure as soon as it enters a new TA region. TPD on vehicle is initialized to preload system public parameters PP and all the identities of registered RSUs. Let V_j be a vehicle with identity ID_{V_j}. Supposing that V_j randomly picks $a'_j \in \mathbb{Z}_q^*$ at time t_j, then its authentication key is $a_j = H_4(a'_j, t_j) \in \mathbb{Z}_q^*$. Vehicle V_j computes $H_j = H_1(ID_{V_j}, a_j)$, $A_j = g_1^{a_j}$ and submits (ID_{V_j}, H_j, A_j) to TA. Then, TA picks $r_j \in \mathbb{Z}_q^*$ and generates challenge $R_j \longleftarrow g_1^{r_j}$ and dynamic password $P_j = A_j^{r_j}$ for V_j. Authentication key a_j and challenge R_j are saved in TPD on vehicle V_j. Meanwhile, TA inserts tuple $\{ID_{V_j}, H_j, A_j, r_j, P_j, T_j\}$ into list $\mathscr{L}$ and tuple $\{A_j, P_j, T_j\}$ into list $\mathscr{L}_{pub}$, where T_j is the expiration date of these two tuples. When tuples in both lists have expired, TA forces corresponding vehicles to update their authentication keys.

5.3. Mutual Authentication. Mutual authentication between vehicle V_j and RSU U_i happens when vehicle V_j is in the transmission range of RSU U_i but does not possess its latest master key. The whole process consists of two stages.

R2V authentication: RSU U_i broadcasts message $(ID_{U_i}, PP_{U_i}, R_i, E_i, t_i, \sigma_{U_i})$ periodically to authenticate itself to newly entered vehicles, where timestamp t_i provides freshness, challenge $R_i = g_1^{r_i}$ that changes along with t_i is used to authenticate vehicle in next stage, element $E_i = g_2^{e_i}$ is used to negotiate symmetric keys with vehicles (both $r_i, e_i \in \mathbb{Z}_q^*$ are picked by RSU U_i and kept secret), and $\sigma_{U_i} \longleftarrow \text{Sign}(sk_{U_i}, M_{R2V})$ is the signature of $M_{R2V} = (ID_{U_i}, E_i, t_i) \in \{0,1\}^*$. After receiving the broadcast message, vehicle V_j first checks whether identity ID_{U_i} has been preloaded into TPD at setup stage or not. If not, vehicle V_j aborts this authentication; otherwise, it verifies signature σ_{U_i} by running $\text{Verify}(PP_s, ID_{U_i}, M_{R2V}, \sigma_{U_i})$. If algorithm Verify outputs "reject", vehicle V_j aborts; otherwise, this authentication succeeds.

V2R authentication: to authenticate itself to RSU U_i, vehicle V_j has to recover its dynamic password P_j in list $\mathscr{L}$ and answer the challenge R_i of RSU U_i with authenticate key a_j.

(1) Vehicle V_j computes $P_j = R_j^{a_j}$ and $P_i = R_i^{a_j}$. Then, it picks $f_j \in \mathbb{Z}_q^*$ and computes $F_j = g_2^{f_j}$ and $K = H_2(E_i^{f_j})$. Key K is used to encrypt $M_{VR} = (P_j, H_3(ID_{U_i}, P_j, P_i, F_j, t_j))$ with algorithm Enc, where t_j is the timestamp. Let $C_j = \text{Enc}(K, M_{VR})$ be the ciphertext of M_{VR}, vehicle V_j replies to RSU U_i with message $(ID_{U_i}, F_j, t_j, C_j)$.

(2) RSU U_i first computes symmetric key $K = H_2(F_j^{e_i})$ and decrypts C_j with K. Supposing that $M'_{VR} = (P'_j, P'_i, H'_{VR})$ is the output of $\text{Dec}(K, C_j)$, if $H'_{VR} \neq H_3(ID_{U_i}, P'_j, P'_i, F_j, t_j)$, RSU U_i aborts; otherwise, RSU U_i searches list $\mathscr{L}_{pub}$ for tuple $\{A', P', T'\}$, where $P' = P'_j$. If such tuple does not exists or has expired, or more than one tuple is found in list $\mathscr{L}_{pub}$, RSU U_i aborts; otherwise, it computes $P''_i = (A')^{r_i}$. If $P''_i = P'_i$, then vehicle V_j manages to authenticate itself to RSU U_i without revealing its real identity.

(3) RSU U_i sends its master key msk_{U_i} to vehicle V_j in ciphertext format $(ID_{U_i}, \widetilde{T}_i, \widetilde{t}_i, \widetilde{C}_i)$, where $\widetilde{T}_i$ is the expiry time of msk_{U_i}, $\widetilde{t}_i$ is a timestamp, and $\widetilde{C}_i = \text{Enc}(K, M_{RV})$, $M_{RV} = (msk_{U_i}, H_3(ID_{U_i}, msk_{U_i}, \widetilde{T}_i, \widetilde{t}_i))$.

(4) Vehicle V_j decrypts C_i and gets $M'_{RV} = (msk'_{U_i}, H'_{RV})$. If $H'_{RV} \neq H_3(ID_{U_i}, msk'_{U_i}, \widetilde{T}_i, \widetilde{t}_i)$, vehicle V_j aborts; otherwise, it stores master key msk'_{U_i} into TPD. Note that this master key will be erased automatically at time $\widetilde{T}'_i$.

5.4. Pseudo-Identity Generation. In terms of privacy preservation, instead of real identities of vehicles, pseudo-identities are generated by TPD to hide the real-world identities of vehicles. Considering a vehicle V_j with real identity ID_{V_j} in the transmission range of RSU U_i, we define its pseudo-identity as $PID_{V_j} = (S, \Pi_0, \Pi_1) = (g_1^s, H_1(ID_{V_j}, a_j)X^s, Y^s Z^{\theta s})$, where $s \in \mathbb{Z}_q^*$ is randomly picked by TPD and $\theta = H_4(g_1^s, H_1(ID_{V_j}, a_j)X^s)$.

We remark that the computation of pseudo-identity of vehicle V_j can be viewed as encrypting $H_1(ID_{V_j}, a_j)$ using Cramer-Shoup encryption scheme (CS scheme) [26] which is secure against adaptive chosen-ciphertext attack (CCA2 secure). The main advantage of such pseudo-identity is that TA could trace the real identity of vehicle by decrypting pseudo-identity. Besides, the nonmalleability of CS scheme does not allow anyone to derive a new and valid pseudo-identity from given one. Using CS scheme might be time-consuming for devices on vehicle, while this problem can be overcame by preparing sufficient pseudo-identities offline in storage device as the on-board storage capacity of vehicle could be extensive.

5.5. Message Signing and Verification. When vehicle V_j locates in the region of RSU U_i, before signing message, it first generates the private key psk_{V_j} of its pseudo-identity PID_{V_j} with master key msk_{U_i} and then signs message m with psk_{V_j} and broadcasts $(PID_{V_j}, ID_{U_i}, m, t_j, \sigma_{V_j})$ to RSUs or vehicles

around, where t_j is a timestamp and $\sigma_{V_j} \longleftarrow \text{Sign}(psk_{V_j}, M)$, $M = (PID_{V_j}, m, t_j)$.

Message m can be verified by running $\text{Verify}(PP_{U_i}, PID_{V_j}, M, \sigma_{V_j})$. However, for verifier that is not in the region of RSU U_i, it has to request the public parameters PP_{U_i} of U_i from nearby RSU. Since IBS scheme $\mathcal{IBS}$ supports the batch verification of multiple signatures, the verifier is able to take advantage of this property to improve the performance of message verification.

5.6. Vehicle Tracing. Pseudo-identity protects the privacy of vehicles on the one hand and facilitates some malicious vehicles to disseminate bogus information on the other. Thus, it is of importance to track down the real identities of misbehaving vehicles which can only be done by TA. Particularly, let PID_{V_j} be one pseudo-identity of malicious vehicle V_j, TA parses PID_{V_j} into (S, Π_0, Π_1) and computes $\theta = H_4(S, \Pi_0)$, $\Pi_1' = S^{y+\theta z}$. If $\Pi_1' \neq \Pi_1$, then this pseudo-identity is invalid; otherwise TA computes $H = \Pi_0 / S^x$. If there exists one valid tuple $\{ID_{V_j}, H_j, A_j, r_j, P_j, T_j\}$ in list $\mathcal{L}$ with $H_j = H$, then TA succeeds to find out the real identity ID_{V_j} of vehicle V_j.

5.7. Secret Parameters Update. There are two secret parameters in TPD that need to be updated regularly: authentication key and RSU's master key. Note that RSU's master key is updated along with V2R authentication. Here, we focus on authentication key update.

(1) Assuming that tuple $\{ID_{V_j}, H_j, A_j, r_j, P_j, T_j\}$ reaches the expiration date T_j, TA generates a pseudo-identity $PID_A = (A_j, H_j A_j^x, A_j^{y+\theta z})$ for vehicle V_j, where $\theta = H_4(A_j, H_j A_j^x)$. Then, TA picks $\hat{r}, r^*, e \in \mathbb{Z}_q^*$ and computes $\widehat{R} = g_1^{\hat{r}}$, $R^* = g_1^{r^*}$, $E = g_2^e$ where challenge $\widehat{R}$ is a test for target vehicle V_j, R^* is a new challenge for V_j and E is used to negotiate key. TA then computes signature $\sigma_A \longleftarrow \text{Sign}(psk_A, M_{TA})$, where $psk_A = \text{Extract}(msk_s, PID_A)$ and $M_{TA} = (PID_A, \widehat{R}, R^*, \widehat{t})$ and broadcasts (M_{TA}, σ_A), where $\widehat{t}$ is a timestamp.

(2) Vehicle V with real identity ID_V and authentication key a that receives this message of TA would check whether $PID_A = (g_1^a, H_1(ID_V, a)X^a, Y^a Z^{\theta a})$, where $\theta = H_4(g_1^a, H_1(ID_V, a)X^a)$. Only V_j that possesses a_j can recognize this pseudo-identity. Then, V_j prepares to update authentication key. It runs $\text{Verify}(PP_s, PID_A, M_{TA}, \sigma_A)$. If signature σ_A is valid and timestamp $\widehat{t}$ is fresh, vehicle V_j picks $a', f \in \mathbb{Z}_q^*$ and computes $a^* = H_4(a', t_j)$, $A^* = g_1^{a^*}$, $H^* = H_1(ID_{V_j}, a^*)$, $F = g_2^f$, $K' = H_2(E^f)$, and $\widehat{P} = \widehat{R}^{a_j}$. Then, vehicle V_j sends message (ID_{TA}, F, t_j, C_j) to TA, where t_j is the timestamp, $C_j = \text{Enc}(K', M_U)$, and $M_U = (A^*, \widehat{P}, H_3(ID_{TA}, A^*, \widehat{P}, t_j))$.

(3) TA recovers $K' = H_2(F^e)$ to decrypt C_j and obtains $M_U' = (A', H', \widehat{P}', H_U')$. If $H_U' \neq H_3(ID_{TA}, A', \widehat{P}', t_j)$, TA aborts; otherwise, it computes $P' = A_j^{\hat{r}}$. If $\widehat{P}' \neq P'$, TA aborts; otherwise, vehicle V_j passes the test of TA; then TA computes $P^* = (A')^{r^*}$ and updates $\{ID_{V_j}, A_j, H_j, r_j, P_j, T_j\}$ with $\{ID_{V_j}, A', H', r^*, P^*, T^*\}$ in list $\mathcal{L}$, where T^* is the expiration time. Also, in list $\mathcal{L}_{pub}$, tuple $\{A_j, P_j, T_j\}$ is updated with $\{A', P^*, T^*\}$. TA picks $\overline{r} \in \mathbb{Z}_q^*$, computes $\overline{R} = g_1^{\overline{r}}$, $\overline{P} = (A')^{\overline{r}}$, and broadcasts $(PID_A, \overline{R}, \overline{P}, \overline{t}, \sigma_A')$, where $\overline{t}$ is a timestamp, $\sigma_A' \longleftarrow \text{Sign}(psk_A, \overline{M}_{TA})$ is the signature of $\overline{M}_{TA} = (PID_A, \overline{P}, \overline{t})$.

(4) Vehicle V_j checks the integrity and validity of message. If signature σ_A' is valid and timestamp $\overline{t}$ is fresh, vehicle V_j computes $\overline{P}' = \overline{R}^{a^*}$. If $\overline{P} \neq \overline{P}'$, vehicle V_j aborts; otherwise, current authentication key a_j and challenge R_j in TPD are replaced with a^* and R^*.

We remark that the centralized update of authentication key might incur DoS attack against TA. Fortunately, there are several effective ways to cope with such attack. First, TA in reality can provide the update service in parallel mode. That is, multiple servers are deployed to interact with vehicles simultaneously which can alleviate the burden on each server and accelerate the overall efficiency. Besides, since TA is the initiator of update procedure, it is able to adaptively adjust the interval of update according to practical situation without compromising the security of whole system. Also, if TA does not receive the reply of one vehicle within a period of time, it would abort the update process with this vehicle and refuse to interact with it temporarily.

6. Security Analysis

This section gives a comprehensive security analysis of our framework. We show that our framework has achieved all the security objectives mentioned in Section 3.

Authentication: one can notice that message authentication is guaranteed by IBS scheme immediately, so we mainly analyze the mutual authentication between vehicle and RSU. In R2V authentication, the generation of signature of broadcasted message needs RSU's private key which is provided by TA. If received signature can be successfully verified with the identity of RSU, vehicle is convinced that current RSU is the sender of messages from the unforgeability of IBS scheme. In V2R authentication, for vehicle V_j, it proves to RSU that it can recover the dynamic password $P_j = A_j^{r_j}$ of tuple $\{A_j, P_j, T_j\}$ in list $\mathcal{L}_{pub}$ and answer the dynamic password $P_i = A_j^{r_i}$ which is corresponding to new challenge $R_i = g_1^{r_i}$ generated by RSU. We claim that given A_j and R_i, other entities that do not possess the authentication key a_j or r_i picked by RSU cannot compute the correct P_i if CDH problem is hard. Therefore, vehicles that send correct dynamic password pair (P_j, P_i) can authenticate themselves

to RSU. Since tuple $\{A_j, P_j, T_j\}$ and P_i are independent of the real identity of V_j, the whole process of authentication does not leak any information about vehicle's identity.

Nonrepudiation: the pseudo-identity of vehicle, corresponding to private key and signature of message broadcasted by vehicle are all generated in TPD. Since pseudo-identity, signature, and timestamp are key components of message, a vehicle cannot deny its behavior of generating message via TPD at certain time. Moreover, the generation of pseudo-identity requires authentication key a_j which is only accessible to vehicle itself. Due to the nonmalleability of CS scheme, we note that one cannot derive a new valid pseudo-identity from given one.

Identity privacy preserving: the pseudo-identity of vehicle V_j is a ciphertext of $H_1(ID_{V_j}, a_j)$ in CS scheme. From the security of this encryption scheme, pseudo-identity does not leak any information about vehicle's real identity. Moreover, the mutual authentication between vehicle and RSU does not leak real identity as well.

Conditional Traceability: the process of tracing vehicle has been described in Section 5 already. Only TA that possesses the private key x, y, z is able to verify the validity of pseudo-identity, recover $H_j = H_1(ID_{V_j}, a_j)$, and find the real identity ID_{V_j} in private list $\mathscr{L}$.

Nonframeability: since vehicle's authentication key is only accessible to itself, TA cannot authenticate itself to RSU as a valid vehicle and obtain the RSU's master key, let alone generating the private keys of pseudo-identities and forging the signatures of vehicles. On the other hand, although RSU owns master key, it cannot generate vehicle's new pseudo-identities and valid signatures as the authentication key is required and collected pseudo-identities do not provide any useful information for pseudo-identity generation. Moreover, although RSU could collect a set of pseudo-identities of vehicles, due to unlinkability, it is impossible for RSU to distinguish certain vehicle's pseudo-identities and to forge serial signatures of this vehicle. TA is also able to detect the reuse of pseudo-identities by decrypting them and querying recovered hash values in maintained list. If TA does not find them in list, then there exists the abuse of pseudo-identities.

Ideal TPD freshness: one can note that secrets in TPD are vehicle's authentication key and RSU's master key. TA is responsible for the update of vehicle's authentication key and RSU would regularly update its master key. Thus, realistic TPD is secure enough to store these secrets and ideal TPD is not needed.

Unlinkability: in our framework, all messages of vehicle are signed with different pseudo-identities which are independent from each other. It is impossible to distinguish whether two random messages are sent by one vehicle or not. Thus, our framework satisfies unlinkability.

Message confidentiality: in V2R authentication, RSU sends its master key in ciphertext form to vehicles that complete current authentication. The master key of RSU is encrypted using symmetric encryption scheme and the negotiation of symmetric key follows the method of Diffie-Hellman key exchange. Thus, transmission of RSU's master key is confidential and secure. Similarly, the same symmetric encryption scheme and key exchange method are applied in transmitting new authentication information of vehicle during updating secret parameters.

Attacks resistance: In proposed framework, we assume that the whole system is initialized in a secure environment, but mutual authentication, message signing and verification, and secret parameter update might suffer various attacks from adversary. We now demonstrate that our framework is resistant to following attacks.

(i) Replay attack: every transmitted message is marked with timestamp. The receiver of message would check the freshness of message via timestamp and discard replayed messages.

(ii) Impersonation attack: in mutual authentication, adversary might try to imitate a valid RSU U_i and gain the trust of vehicles. However, the private key of RSU U_i is generated in initialization phase and securely kept by RSU U_i. Adversary cannot access to this private key. Thus, the signature of its message cannot be verified with the identity of U_i. In secret parameters update, if adversary (e.g., registered vehicles or RSUs) wants to impersonate the TA and send update instruction to vehicle V_j, it has to compute the special pseudo-identity PID_A of V_j with public parameters X, Y, Z. The hardness of computing $PID_A = (A_j, H_j A_j^x, A_j^{y+\theta z})$ with A_j and X, Y, Z can be reduced to CDH assumption. One update instruction is targeted at only one vehicle, but other irrelevant vehicles might also receive this instruction. If a malicious vehicle intends to imitate the target one V_j, it has to answer the challenge $\widehat{R}$ of TA with dynamic password $\widehat{P}$. Given $\widehat{R}$ and A_j, it is still hard to compute $\widehat{P} = \widehat{R}^{a_j} = A_j^{\widehat{r}}$ according to CDH assumption. Thus, our framework could withstand impersonation attack.

(iii) Modification attack: for signed message, making any modifications could result in the failure of verification from the correctness of IBS scheme. For encrypted message, the plaintext and its hash value are concatenated and encrypted together. If ciphertext is modified arbitrarily, the underlying plaintext cannot be verified with its hash value.

(iv) Side-channel attack: this attack is mainly for vehicle's TPD which stores sensitive data including authentication key and master key of RSU. It is worth mentioning that the real identity of vehicle is not stored in TPD. In our protocol, these secret parameters are updated frequently such that adversary cannot obtain sufficient data through side-channel analysis. Moreover, new secret parameters are independent of the old ones, so the leakage of old parameters does not benefit the guessing of new ones.

Remarks: it is worth mentioning that Sybil attack is inevitable and ubiquitous in most of cryptographic schemes and thus the detection of such attack has been extensively studied

Table 1: Notations of different execution time.

Notation	Description
T_m	Average execution time of multiplication operation on group $\mathbb{G}_1$ or $\mathbb{G}_2$.
T_a	Execution time of addition operation on group $\mathbb{G}_1$ or $\mathbb{G}_2$.
T_p	Execution time of bilinear pairing operation $e(\cdot,\cdot)$.
T_{pm}	Average execution time of multiplication operation on group $\mathbb{G}$.
T_{pa}	Execution time of addition operation on group $\mathbb{G}$.
T_h	Average execution time of general hash operation.
$T_{sign}^{\mathcal{IBS}}$	Execution time of signing message using $\mathcal{IBS}$ scheme.
$T_{vrfy}^{\mathcal{IBS}}$	Execution time of verifying signature using $\mathcal{IBS}$ scheme.
T_{aes}	Execution time of AES encryption / decryption.
T_{query}	Execution time of searching list.

[27]. Our framework is vulnerable to Sybil attack as any authenticated vehicle is accessible to the master key of RSU and can imitate other vehicles at the same time by forging their messages. However, we claim that it is possible to detect such attack by authorities in proposed framework. Precisely, misbehaving vehicle cannot imitate other vehicles as it does not know their authentication keys and cannot generate correct pseudo-identities. Collected pseudo-identities also do not help in computing new ones from the nonmalleability of CS scheme. Besides, the reuse of collected pseudo-identities can be detected by TA. Consequently, only pseudo-identities of misbehaving vehicle itself can be generated to conduct Sybil attack. The TA is able to detect such attack easily by revealing its real identity from pseudo-identities.

7. Performance Analysis

In this section, we evaluate both the computation and communication costs of authentication in our framework and make a comparison with existing works. To achieve 80-bit security, elliptic curve groups $\mathbb{G}_1, \mathbb{G}_2$ with 160-bit prime order q, IBS scheme $\mathcal{IBS}$, and symmetric encryption scheme AES with 80-bit security are used in our protocol. In pairing-based IBS scheme $\mathcal{IBS}$, we use bilinear pairing e: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ to realize 80-bit security level, where $\mathbb{G}$ is an additive group with 160-bit prime order p on supersingular elliptic curve with embedding degree 2. The sizes of elements in $\mathbb{G}_1$ and $\mathbb{G}$ are 320 bits and 1024 bits. For the convenience of discussion, notations of execution time are defined in Table 1.

According to [6], bilinear pairing operation that takes $T_p \approx 4ms$ is the most time-consuming operation. Other bilinear pairing-related operations cost more time than corresponding operations in ECC. That is, $T_{pm} \approx 3.9T_m = 1.7ms$ and $T_{pa} \approx 3.9T_a$. The execution time of multiplication operation T_m/T_{pm} is approximately 240 times greater than T_a/T_{pa}. In comparison to above execution time, T_h, T_{aes}, and T_{query} could be negligible.

7.1. Comparison of Different IBS Schemes. IBOOS and OT-IBS schemes usually might be more efficient than traditional IBS schemes. In IBOOS scheme, time-consuming operations can be completed in offline stage, and the actual signing time is determined by online stage. The structure of OT-IBS scheme is commonly much simpler than that of traditional IBS scheme because of the one-time usage of private key. Moreover, IBS schemes that support batch verification could amortize time-consuming operation over a bundle of signatures. Therefore, for better performance, we only investigate existing IBOOS and OT-IBS schemes that support batch verification.

Table 2 shows the comparison of signing time T_{sign}, verification time T_{vrfy}, and signature size. One can note that bilinear pairing-based IBS schemes XMS and ZWD are less efficient than ECC-based IBS schemes in both verification time and signature size. Schemes LBZ and HZS have same verification time. However, the signing time of LBZ is correlated with the bit length n of message, so it might be greater than that of HZS for long messages. Moreover, scheme HZS enjoys the shortest signature among these schemes. Thus, in following discussion, we adopt scheme HZS as the IBS scheme in our framework.

7.2. Authentication Efficiency. When evaluating an authentication protocol, we are most concerned about the time and communication costs of authentication. In our framework, vehicle that just enters a new RSU region has to complete the mutual authentication with RSU in time; otherwise it cannot communicate with other entities. Thus, we consider the overhead of mutual authentication from the perspective of vehicle. Since RSU broadcasts messages periodically, it is reasonable to assume that vehicle receives these messages as soon as it drives into the region of RSU. The computational overhead of R2V authentication is mainly determined by $T_{vrfy}^{HZS} = 3T_m + 2T_a$. Before replying to RSU, vehicle has to spend time $T_{gen}^{v} = 3T_m + 2T_h + T_{aes}$ to generate message $(ID_{U_i}, F_j, t_j, C_j)$. In AES, the size of ciphertext is the same as plaintext, so ciphertext C_j is 320∗2+160=800-bit long. Suppose that the size of identity of RSU ID_{U_i} is 160 bits, timestamps are 20 bits, and the length of message $(ID_{U_i}, F_j, t_j, C_j)$ is 160+320+20+800=1300 bits. Then, RSU spends time $T_{proc}^{r} = 2T_m + 3T_h + 2T_{aes} + T_{query}$ to process the message from vehicle and prepare master key for it if authentication succeeds, where T_{query} is the execution time of searching list $\mathcal{L}_{pub}$. The length of message $(ID_{U_i}, \widetilde{T}_i, \widetilde{t}_i, \widetilde{C}_i)$ sent by RSU is 160+20+20+(160+160)=520 bits. Vehicle needs

Table 2: Efficiency of different IBS schemes.

Signature Type	Scheme	Signing Time T_{sign}	Verification Time T_{vrfy}	Signature Size (bits)
IBOOS	XMS [28]	negligible	$2T_p + 2T_{pm} + T_{pa}$	$2\lvert\mathbb{G}\rvert + \lvert p\rvert \approx 2208$
	LBZ [29]	nT_a	$3T_m + 2T_a$	$2\lvert\mathbb{G}_1\rvert + \lvert q\rvert \approx 800$
OT-IBS	HZS [6]	T_m	$3T_m + 2T_a$	$\lvert\mathbb{G}_1\rvert + \lvert q\rvert \approx 480$
	ZWD [13]	$T_{pm} + T_{pa}$	$2T_p + T_{pm} + T_{pa}$	$\lvert\mathbb{G}\rvert + \lvert p\rvert \approx 1184$

Table 3: Comparison of mutual authentication efficiency.

Mutual Authentication	Computation Overhead	Communication Overhead (bits)
Li et al.[7]	$9T_m + 726T_a \approx 2886T_a$	3200
Zhang et al.[13]	$2T_p + 6T_{pm} + T_{pa} \approx 10300T_a$	3268
Our framework	$8T_m + 2T_a \approx 1920T_a$	1820

time $T^v_{dec} = T_{aes} + T_h$ to decrypt it and check the master key of RSU. Therefore, the overall computation overhead of mutual authentication is $T_{v\longleftrightarrow r} = T^{HZS}_{vrfy} + T^v_{gen} + T^r_{proc} + T^v_{dec} = 8T_m + 2T_a + 6T_h + 4T_{aes} + T_{query} \approx 8T_m + 2T_a$, and the communication overhead is 1300+520=1820 bits.

Similarly, we also analyze the efficiency of mutual authentication in existing works. In [7], LBZ scheme is used to sign messages during mutual authentication. Suppose that the sizes of code HR of vehicle's home region, nonce nc, and join request $join$ are 20 bits and ciphertext E_{pk} of vehicle's real identity is 320 bits; then the length of vehicle's pseudo-identity $T\|E_{pk}\|HR\|ID_U$ is 20+320+20+160=520 bits. Then message $(ID, PS, t, join, Sign(PS, t))$ sent by vehicle is 160+520+20+20+800=1520 bits. The offline signature of LBZ scheme actually could be preloaded by vehicle, so the message $(ID, t, PS, ID_r, nc, Sign(ID_r, t))$ returned from RSU is 160+20+520+160+20+800=1680 bits. The total communication overhead is 3200 bits. The computation overhead is $3T^{LBZ}_{vrfy} + T^{LBZ}_{sign} + T'^{LBZ}_{sign} = 3 * (3T_m + 2T_a) + 540T_a + 180T_a$, where T^{LBZ}_{sign} is correlated to the length of input.

In [13], the computational overhead of protocol is $T^{ZWD}_{vrfy} + T^v_{gen} + T^r_{proc} + T^v_{dec} = 2T_p + 6T_{pm} + T_{pa} + 6T_h + 4T_{aes} + T_{query} \approx 2T_p + 6T_{pm} + T_{pa}$. Suppose that the length of authentication key λ is 160 bits, element $F \in \mathbb{G}$ is 1024 bits, then message (F, ID, C, t) sent by vehicle is 1024+160+(160+20)+20=1384 bits, where $C = \mathrm{Enc}(\lambda, t)$. RSU returns message (H, C') which is 160+(20+160+160)=500 bits long to vehicle. Since RSU has to forward message sent by vehicle to TA, then the communication overhead is 1384∗2+500=3268 bits.

Table 3 shows the comparison of computation and communication overhead of mutual authentication between vehicle and RSU, where computation overhead is represented as the multiples of T_a. The improvement on computation is $(2886T_a - 1920T_a)/2886T_a \approx 33.5\%$ over [7] and $(10300T_a - 1920T_a)/10300T_a \approx 81.4\%$ over [13]. In communication overhead, there are $(3200 - 1820)/3200 \approx 43.1\%$ and $(3268 - 1820)/3268 \approx 44.3\%$ improvements over [7] and [13], respectively.

7.3. Secret Parameters Update Efficiency. At the beginning of update secret parameters, TA has to compute 3∗320=960-bit long pseudo-identity PID_A first, which costs time $2T_m + T_a$ and then challenges $\widehat{R}, R^*$, element E, and signature $Sign_A$. The overall broadcast message is 960+320+320+320+20+480=2420 bits long and TA spends time $(2T_m + T_a) + 3T_m + T_m = 6T_m + T_a$ to generate it.

We assume that vehicle V_j has computed its PID_A in advance and could recognize the update instruction from TA immediately after receiving the broadcast message. Vehicle V_j first verifies the signature, picks new authentication key, and then responds TA with new authentication information (ID, F, t, C) which is 160+320+20+(320+160+320+160)=1460-bit long. This process would take $T^{HZS}_{vrfy} + 4T_m + 3T_h + T_{aes} \approx 7T_m + 2T_a$.

After receiving the authentication information from vehicle V_j, TA needs to spend time $T_{aes} + T_m$ to check its integrity. If received information is complete and valid, TA has to prove to V_j that it possesses the latest information of V_j by broadcasting message $(PID_A, \overline{R}, \overline{P}, t, Sign)$whose length is 960+320+320+20+480=2100 bits. The time of generating message is $2T_m + T^{HZS}_{sign} = 3T_m$.

Finally, vehicle V_j takes $T^{HZS}_{vrfy} + T_m = 4T_m + 2T_a$ to verify the message sent by TA, and the procedure of secret parameters update is finished. Overall, the time cost on the vehicle side is $(7T_m + 2T_a) + (4T_m + 2T_a) = 11T_m + 4T_a$, and $(6T_m + T_a) + 3T_m = 9T_m + T_a$ on the TA side. The communication costs are 1460 bits and 2420+2100=4520 bits for vehicle and TA, respectively.

8. Conclusions

In this paper, we propose a practical framework of CPPA scheme that does not rely on ideal TPD and supports realistic TPD. This feature makes our framework more suitable for practical use. In addition to traditional security requirements, such as nonrepudiation and conditional privacy preservation, our framework also achieves nonframeability that prevents TA and RSUs from framing innocent vehicles. Performance analysis shows that our framework outperforms existing schemes in terms of mutual authentication.

Data Availability

The data of execution time supporting the findings of this study are from previously reported studies, which have been cited.

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Key R&D Program of China under grants 2017YFB0802300, the National Natural Science Foundation of China [61702541, 61872087], the Young Elite Scientists Sponsorship Program by CAST [2017QNRC001], and the Science Research Plan Program by NUDT [ZK17-03-46].

References

[1] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[2] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 246–250, IEEE, Phoenix, Ariz, USA, April 2008.

[3] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: a robust signature scheme for vehicular networks using binary authentication tree," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1974–1983, 2009.

[4] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1227–1239, 2010.

[5] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "ABAKA: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 248–262, 2011.

[6] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[7] J. Li, H. Lu, and M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2015.

[8] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896–911, 2016.

[9] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[10] J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," in *Proceedings of the International Conference on Research in Smart Cards*, pp. 200–210, Springer, Berlin, Germany, 2001.

[11] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the Annual International Cryptology Conference*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer, Berlin, Germany, 1999.

[12] C. Zhang, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3357–3368, 2008.

[13] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 516–526, 2016.

[14] S.-J. Horng, S.-F. Tzeng, Y. Pan et al., "B-SPECS+: batch verification for secure pseudonymous authentication in VANET," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1860–1875, 2013.

[15] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1229–1237, IEEE, April 2008.

[16] C. Lyu, D. Gu, Y. Zeng, and P. Mohapatra, "PBA: prediction-based authentication for vehicle-to-vehicle communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 71–83, 2016.

[17] J. Shen, T. Zhou, F. Wei, X. Sun, and Y. Xiang, "Privacy-preserving and lightweight key agreement protocol for v2g in the social internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2526–2536, 2018.

[18] A. Perrig, R. Canetti, D. J. Tygar et al., "The TESLA broadcast authentication protocol," *Rsa Cryptobytes*, vol. 5, no. 2, pp. 2–13, 2002.

[19] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing together efficient authentication, revocation, and privacy in VANETs," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2009*, pp. 1–9, IEEE, Italy, June 2009.

[20] Q. Wu, J. Domingo-Ferrer, and Ú. González-Nicolás, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 559–573, 2010.

[21] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A scalable robust authentication protocol for secure vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1606–1617, 2010.

[22] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 907–919, 2014.

[23] K.-A. Shim, "CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1874–1883, 2012.

[24] X. Lin and X. Li, "Achieving efficient cooperative message authentication in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3339–3348, 2013.

[25] J. Liu, Y. Yu, Y. Zhao et al., "An efficient privacy preserving batch authentication scheme with deterable function for VANETs," in *Proceedings of the International Conference on Network and System Security*, pp. 288–303, Cham, Swizerland, 2018.

[26] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, L. R. Knudsen, Ed., vol. 2332 of *Lecture Notes in Computer Science*, pp. 45–64, Springer, Berlin, Germany, 2002.

[27] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[28] S. Xu, Y. Mu, and W. Susilo, "Efficient authentication scheme for routing in mobile ad hoc networks," in *Proceedings of the International Conference on Embedded and Ubiquitous Computing*, vol. 3823 of *Lecture Notes in Computer Science*, pp. 854–863, Springer, Berlin, Germany, 2005.

[29] J. K. Liu, J. Baek, J. Zhou, Y. Yang, and J. W. Wong, "Efficient online/offline identity-based signature for wireless sensor network," *International Journal of Information Security*, vol. 9, no. 4, pp. 287–296, 2010.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 3249726, 9 pages
https://doi.org/10.1155/2019/3249726

Research Article

Efficient Attribute-Based Encryption with Privacy-Preserving Key Generation and Its Application in Industrial Cloud

Yujiao Song,[1] Hao Wang,[1,2] Xiaochao Wei,[1] and Lei Wu[1,3]

[1]*School of Information Science and Engineering, Shandong Normal University, China*
[2]*School of Computing and Information Technology, University of Wollongong, Australia*
[3]*Shandong Provincial Key Laboratory of Software Engineering, China*

Correspondence should be addressed to Hao Wang; wanghao@sdnu.edu.cn

Received 9 March 2019; Accepted 30 April 2019; Published 23 May 2019

Guest Editor: Mingwu Zhang

Due to the rapid development of new technologies such as cloud computing, Internet of Things (IoT), and mobile Internet, the data volumes are exploding. Particularly, in the industrial field, a large amount of data is generated every day. How to manage and use industrial Big Data primely is a thorny challenge for every industrial enterprise manager. As an emerging form of service, cloud computing technology provides a good solution. It receives more and more attention and support due to its flexible configuration, on-demand purchase, and easy maintenance. Using cloud technology, enterprises get rid of the heavy data management work and concentrate on their main business. Although cloud technology has many advantages, there are still many problems in terms of security and privacy. To protect the confidentiality of the data, the mainstream solution is encrypting data before uploading. In order to achieve flexible access control to encrypted data, attribute-based encryption (ABE) is an outstanding candidate. At present, more and more applications are using ABE to ensure data security. However, the privacy protection issues during the key generation phase are not considered in the current ABE systems. That is to say, the key generation center (KGC) knows both of attributes and corresponding keys of each user. This problem is especially serious in the industrial big data scenario, because it will cause great damage to the business secrets of industrial enterprises. In this paper, we design a new ABE scheme that protects user's privacy during key issuing. In our new scheme, we separate the functionality of attribute auditing and key generating to ensure that the KGC cannot know user's attributes and that the attribute auditing center (AAC) cannot obtain the user's secret key. This is ideal for many privacy-sensitive scenarios, such as industrial big data scenario.

1. Introduction

Due to the rapid development of new technologies such as cloud computing, Internet of Things (IoT), and mobile Internet, the data volumes are exploding, and we have truly entered the era of "Big Data." Big Data technology has been focused and applied to almost every industry, retail, healthcare, financial services, government, and so on. Particularly, in the field of industrial production, a large amount of data is generated every day, and it includes business data from information systems, machine data from industrial IoT systems, and some other data from related websites, etc. For a manufacturing enterprise, Big Data can not only be used to improve the efficiency of the business, but more importantly change the manufacturing process and business model. Industrial Big Data is the core of intelligent manufacturing and industrial IoT and provides the most favorable support for the development of Industry 4.0. How to manage and use industrial Big Data efficiently is a great challenge for every enterprise manager.

Cloud computing technology can provide better solutions to the above challenge. Using cloud technology, enterprises get rid of the heavy data management work and concentrate on their main business. Nowadays, large cloud service providers, such as Amazon, Microsoft, IBM, etc., have launched industrial cloud platforms, and more and more industrial enterprises migrate their data to these platforms. However, hosting data to third-party platforms will create new problems, because the security and privacy of the data have to depend on the credibility of the third-party.

For businesses, the biggest concern is the confidentiality of industrial data. The main solution to this problem is to use encrypting methods to protect data before uploading it. However, traditional symmetric and asymmetric encryption schemes are not appropriate for providing fine-grained access control. Therefore, the above problems have brought new challenges to data encryption, and numerous studies have focused on these issues [1–3].

Among various solutions, attribute-based encryption (ABE) [4, 5] has become an excellent candidate because of its ability to provide data confidentiality and fine-grained access control for cloud storage. Currently, more and more industrial enterprises are using ABE. In an industrial alliance, enterprises can share encrypted data based on the attributes. Only those enterprises whose attributes meet the access policy can decrypt the encrypted data. Although much research has been done on ABE [6–8], there are still some problems that have not been solved well. The current ABE systems do not consider privacy protection during the key generation phase. That is to say, the key generation center (KGC) knows the attributes and corresponding keys of each user in this system. This causes great damage to the user's privacy and data confidentiality. Particularly, in the application scenarios of industrial big data, the attributes of enterprise users may be related to the business secrets of enterprises.

1.1. Our Contribution. In order to solve the privacy protection problem in key generation phase, we propose a new ABE system, in which we separate the functionality of attribute auditing and key extracting to ensure that the KGC does not know the specific attributes of the user and that the attribute auditing center (AAC) does not obtain the user's key. In this system, when user applies its private key, it authenticates its attributes to AAC first and gets a blind token, which only certificates its attributes blindly and reveals nothing about specific attributes. The user presents the blind token to the KGC to obtain the corresponding blind key, from which user can extract the final private key. During this process, no information about the user's attributes is leaked to the KGC, and no information about the private key is leaked to the AAC. We implicitly use the oblivious transfer (OT) protocol to solve this problem. This protects the user's privacy during key generation phase.

Our ABE is suitable for privacy sensitive scenarios. Particularly, in the encryption system of industrial cloud, the attributes often involve business secrets of industrial enterprises. KGC, as a technology department, should not know these types of secret information. Therefore, we expressly introduce an application of our new scheme in the industrial cloud.

1.2. Related Work

1.2.1. Attribute-Based Encryption. Attribute-based encryption is a one-to-many public key encryption. Only the user, whose attributes satisfy the access policy set by the encryptor, can decrypt the ciphertext. This concept originates from identity-based encryption [9]. In 2005, Sahai and Waters [4] proposed the concept of fuzzy identity encryption, which became a precedent for attribute-based encryption. In 2006, Goyal et al. [5] first proposed the formal definition of attribute-based encryption (ABE), which classifies as key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). They also constructed the first KP-ABE scheme. In the next year, Bethencourt et al. [10] gave the CP-ABE construction for the first time. In a CP-ABE scheme, the encryptor sets an access policy in the ciphertext to determine which kind of users can decrypt the data. This is very consistent with the security requirements of cloud storage. In recent years, more and more researches focus on CP-ABE [11–13]. However, none of the aforementioned works deals with privacy protection problem in the key generation phase.

1.2.2. Oblivious Transfer. The concept of oblivious transfer (OT) is originally proposed by Rabin [14] in 1981, and then it became an important basic primitive in the field of cryptography. In an OT protocol, the sender delivers part of messages to the receiver and is still unaware of which parts (if any) are delivered. In other words, a secure OT protocol must satisfy two security features: (1) the sender cannot obtain the selection information of the receiver; (2) the receiver cannot obtain any information about other messages except for its choice.

In 1985, Even et al. [15] presented a specific 1-out-of-2 OT protocol (OT_2^1), in which the sender S has 2 values, and receiver R only gets one of them. Then, Brassard et al. [16] extended OT_2^1 to OT_n^1. In 1998, Stern [17] gave a generalized construction of OT protocol based on public key encryption. In 2001, Naor and Pinkas [18] gave a 2-round OT_n^1 protocol based on Diffie-Hellman assumption without random oracle. In the same year, Aiello et al. [19] constructed a 2-round OT_n^1 protocol based on homomorphic public-key encryption. In 2002, Tzeng [20] gave an OT_n^1 protocol with better round complexity and better communication complexity. In 2003, Ishai et al. [21] proposed OT extension, from which a large number of OTs can be performed using only cheap symmetric-key operations. In the past decades, OT protocol has been fully studied and widely used [22–25].

1.3. Organization. In Section 2, we introduce the preliminaries of this paper. In Section 3, we introduce the concept of attribute-based encryption with privacy preserving key generation (PPKG-ABE) and its security definition. In Section 4, we propose a specific PPKG-ABE scheme and analyze its security in Section 5. In Section 6, we introduce the application of PPKG-ABE in industrial cloud environment for protecting the security of industrial Big Data.

2. Preliminaries

2.1. CP-ABE. In CP-ABE system, there are three types of entities, i.e., key generation center (KGC), encryptor, and decryptor. The KGC issues secret key according to users' attributes. The encryptor encrypts the messages according to a designated access policy. The decryptor can decrypt the ciphertext successfully only if its attributes satisfy the corresponding access policy.

There are four algorithms in a CP-ABE scheme:

(1) Setup: it takes security parameters as input and outputs public parameters PP and master secret key MSK.

(2) KeyGen: it takes public parameters PP, master secret key MSK, and a set of attributes S as input and outputs secret key SK_S corresponding to S.

(3) Encryption: it takes public parameters PP, access policy $\mathbb{W}$, and message M as input and outputs the ciphertext $CT_{\mathbb{W}}$.

(4) Decryption: it takes public parameters PP, ciphertext $CT_{\mathbb{W}}$, and secret key SK_S as input and outputs the message M, if and only if the attributes S satisfy the access policy $\mathbb{W}$; i.e., $S \vDash \mathbb{W}$.

2.2. Oblivious Transfer. The oblivious transfer (OT) protocol is a two-party computation protocol in which one party is the sender ($\mathcal{S}$) and the other is the recipient ($\mathcal{R}$). The protocol ensures the following: $\mathcal{S}$ sends a group of messages to $\mathcal{R}$. $\mathcal{R}$ can get a subset of these messages, but $\mathcal{S}$ does not know which messages that $\mathcal{R}$ received.

In this paper, we draw on a classic (OT_2^1) protocol [26]:

Party $\mathcal{S}$ has two elements δ_0, δ_1 of group $\mathbb{G}$ and party $\mathcal{R}$ has a bit $b \in \{0,1\}$. The descriptions of group $\mathbb{G}$ are known to both parties, where $|\mathbb{G}| = q$ and g is a generator.

(1) $\mathcal{R}$ randomly chooses $\alpha, \beta, \gamma \in [1,q]$ and sets τ as follows:

(a) If $\sigma = 0$, then $\tau = (g^\alpha, g^\beta, g^{\alpha\beta}, g^\gamma)$.

(b) If $\sigma = 1$, then $\tau = (g^\alpha, g^\beta, g^\gamma, g^{\alpha\beta})$.

$\mathcal{R}$ sends τ to $\mathcal{S}$.

(2) $\mathcal{S}$ receives $\tau = (x, y, z_0, z_1)$. Then, $\mathcal{S}$ checks $z_0 \neq z_1$. If not, it outputs $\perp$, and aborts.

In addition, $\mathcal{S}$ chooses $u_0, u_1, v_0, v_1 \in [1,q]$ randomly and computes the following 4 values:

$$\begin{aligned} \omega_0 &= x^{u_0} \cdot g^{v_0}, \\ k_0 &= (z_0)^{u_0} \cdot y^{v_0} \\ \omega_1 &= x^{u_1} \cdot g^{v_1}, \\ k_1 &= (z_1)^{u_1} \cdot y^{v_1} \end{aligned} \tag{1}$$

Then, $\mathcal{S}$ calculates $c_0 = x_0 \cdot k_0, c_1 = x_1 \cdot k_1$ and sends (ω_0, c_0) and (ω_1, c_1) to $\mathcal{R}$.

Finally, $\mathcal{R}$ calculates $k_\sigma = (\omega_0)^\beta$ and obtains $\delta_\sigma = c_\sigma \cdot (k_\sigma)^{-1}$.

2.3. Bilinear Maps. Let $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ be three q order cyclic groups. The bilinear pairing operation e is a bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, and satisfies the following properties:

(1) $\forall g \in \mathbb{G}_1, \forall h \in \mathbb{G}_2, \forall x, y \in Z_q^*$, there is $e(g^x, h^y) = e(g,h)^{xy}$(2) $\exists g_0 \in \mathbb{G}_1, \exists h_0 \in \mathbb{G}_2, e(g_0, h_0) \neq 1$(3) $\forall g \in \mathbb{G}_1$, $\forall h \in \mathbb{G}_2, e(g,h)$ can be computed in polynomial time

In this paper, we use asymmetric bilinear groups; that is, $\mathbb{G}_1 \neq \mathbb{G}_2$.

2.4. Security Assumption

Definition 1. Let $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ form bilinear groups, let g be a generator of $\mathbb{G}_1$, and let h be a generator of $\mathbb{G}_2$. For some unknown $\alpha \in \mathbb{Z}_p^*$, define $g_i = g^{\alpha^i}$, and set $\overrightarrow{y}_{g,\alpha,n} = (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$. We say an algorithm $\mathcal{B}$ solves n-BDHE problem with advantage ϵ, if on input $g, h, \overrightarrow{y}_{g,\alpha,n}$,

$$\left| Pr\left[\mathcal{B}\left(e\left(g_{n+1}, h\right)\right) = 1\right] - Pr\left[\mathcal{B}\left(Z\right) = 1\right]\right| \geq \epsilon, \tag{2}$$

where Z is a random element of $\mathbb{G}_T^*$.

The decision n-BDHE assumption holds if ϵ is negligible for any polynomial algorithm.

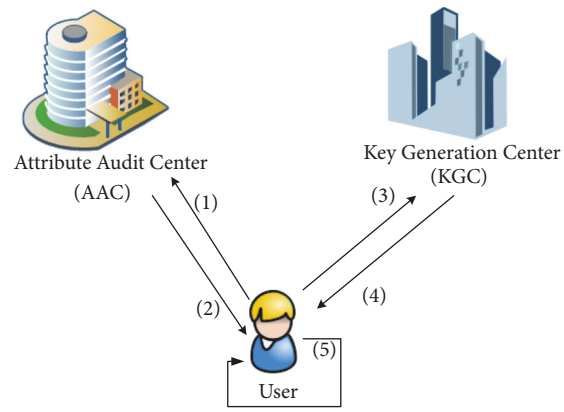


Figure 1: System model.

3. Attribute-Based Encryption with Privacy Preserving Key Generation

In the key generation phase of traditional ABE, KGC always knows the attribute information of each user. This has greatly damaged the privacy of users. In order to solve this problem, we separate the two functions of attribute auditing and key extracting. We introduce an attribute audit center (AAC) in ABE system to authenticate the attributes of users and to make blind token for them. KGC, as a simple technical support institution, is only responsible for generating keys, but it does not know the corresponding attributes of these keys.

3.1. System Model. In the key generation phase (as shown in Figure 1), there are three types of entities: attribute audit center (AAC), key generation center (KGC), and data user. In this system, user submits its attributes and relevant evidence to AAC. The AAC audits the user's attributes and returns a blind token with the signature of AAC to user. In practical applications, AAC is often carried out by the institutions that provide certification for user's attributes, such as government offices, because they know the attributes of users themselves and do not cause extra leaks. In other words, the blind token is the evidence for users owning some attributes. This token does not reveal any information of user's attributes and only ensures the authenticity. When user needs to obtain its attributes key, it will submit the blind token to KGC, which is a technical institution. The KGC first checks the legitimacy of the token; if the token is invalid, it aborts; otherwise, it runs the key generation algorithm on the token and returns a blind

key. After user obtains the blind key, it extracts the secret key locally.

The specific process is as follows:

(1) The user shows its attributes and relevant evidence to the attribute audit center (AAC).

(2) The AAC audits the user's attributes and returns a blind token to the user with its signature.

(3) When a user needs to obtain its attributes key, it will submit its blind token to the key generation center (KGC). The KGC cannot get any information about the user's attributes. It only can confirm that the user truly has related attributes.

(4) The key generation center (KGC) first checks the legitimacy of the token, and if the signature is illegal, it aborts; otherwise, it runs the key generation algorithm and outputs a blind key.

(5) The user receives the blind key from KGC and extracts the private key.

3.2. Syntax. In detail, an attribute-based encryption with privacy preserving key generation scheme (PPKG-ABE) includes seven fundamental algorithms: *Setup*, *UserTemKeyGen*, *BlindTokenGen*, *BlindKenGen*, *KeyExtra*, *Encrypt*, and *Decrypt*. The specific algorithms are described as follows:

Setup(κ)$\longrightarrow$ PP, MK: the setup algorithm is run by KGC, it inputs security parameter κ, and it outputs public parameters PP and master secret key MSK.

UserTemKeyGen(PP, κ)$\longrightarrow$ TPK_{User}, TSK_{User}: the user's temporary-key generation algorithm is run by user. It takes PP and security parameters κ as input and outputs user's temporary public key TPK_{User} and user's temporary secret key TSK_{User}.

BlindTokenGen(PP, S, TPK_{User})$\longrightarrow$ T_S: the blind token generation algorithm is run by AAC. It takes PP, user's attributes set S, and user's temporary public key TPK_{User} as input and outputs a blind token T for attributes set S.

BlindKenGen(PP, MSK, T_S)$\longrightarrow$ BSK: the blind key generation algorithm is run by KGC. It takes PP, master secret key MSK, and user's blind token T_S as input and outputs blind secret key BSK for attributes set S.

KeyExtra(BSK_S, TSK_{User})$\longrightarrow$ SK_S: the key extract algorithm is run by user locally. It takes blind secret key BSK_S and user's temporary secret key TSK_{User} as input and outputs the final secret key SK for attributes set S.

Encrypt(PP, M, $\mathbb{W}$)$\longrightarrow$ CT: the encryption algorithm is run by encryptor. It takes PP, message M, and access structure $\mathbb{W}$ as input and outputs ciphertext CT.

Decrypt($CT_{\mathbb{W}}$, SK_S)$\longrightarrow$ M: the decryption algorithm is run by decryptor. It takes ciphertext $CT_{\mathbb{W}}$ and secret key SK_S as input and outputs message M, if $S \vDash \mathbb{W}$.

We note, in PPKG-ABE scheme, that AAC is responsible for auditing user's attributes and issuing blind token T_S to user. The blind token includes a description of the authenticity of user's attributes, along with the signature of AAC, and reveals on information about specific attributes.

3.3. Security Model. We define the security in two aspects: confidentiality and privacy. Specifically, in this security model, we do not allow AAC and KGC to collude.

3.3.1. Confidentiality. We introduce the selective security model of choosing plaintext attacks for the PPKG-ABE scheme. The specific process is working between adversary $\mathscr{A}$ and challenger $\mathscr{C}$:

Init. $\mathscr{A}$ specifies an access structure $\mathbb{W}^*$ for challenge.

Setup. $\mathscr{C}$ calls the Setup algorithm and returns PP to $\mathscr{A}$.

Phase 2. $\mathscr{A}$ queries secret key on any attributes set $S \nvDash \mathbb{W}^*$. $\mathscr{C}$ returns the secret key SK for S.

Challenge. $\mathscr{A}$ submits two messages M_0^* and M_1^*, where $|M_0^*| = |M_1^*|$. $\mathscr{C}$ chooses $b \in 0, 1$ randomly and encrypts M_b^* under $\mathbb{W}^*$. Then, it returns CT^* to $\mathscr{A}$.

Phase 3. Repeats as Phase 2.

Guess. $\mathscr{A}$ guesses b' for b. The advantage Adv for $\mathscr{A}$ is defined as $Pr[b' = b]$ - 1/2.

Definition 4. The PPKG-ABE scheme is selectively IND-CPA secure if Adv is negligible for any polynomial time adversaries.

3.3.2. Privacy. We introduce a new security game for defining privacy. In this game, we define the following two oracles.

Blind Token Oracle $\mathcal{O}_{BT}$(S): it takes attributes set S as input and outputs corresponding blind token T_S.

Blind Key Oracle $\mathcal{O}_{BK}(T_S)$: it takes blind token T_S as input and outputs corresponding blind key BSK_S.

The specific process is working between adversary $\mathscr{A}$ and challenger $\mathscr{C}$:

Setup. $\mathscr{C}$ calls the Setup algorithm and returns PP to $\mathscr{A}$.

Phase 5. $\mathscr{A}$ queries blind token oracle $\mathcal{O}_{BT}$ and blind key oracle freely.

Challenge. $\mathscr{A}$ submits two attributes sets S_0^* and S_1^*, where $|S_0^*| = |S_1^*|$. $\mathscr{C}$ chooses $b \in \{0, 1\}$ randomly and queries blind token oracle $\mathcal{O}_{BT}$ on input S_b^*. It returns blind token $T_{S_b^*}$ to $\mathscr{A}$.

Phase 6. Repeats as Phase 5.

Guess. $\mathscr{A}$ guesses b' for b. The advantage Adv for $\mathscr{A}$ is defined as $Pr[b' = b]$ - 1/2.

Definition 7. The PPKG-ABE scheme is privacy-protected in key generation phase, if Adv is negligible for any polynomial time adversaries.

4. A Specific PPKG-ABE Scheme

4.1. Construction. In this construction, the PPKG-ABE scheme is constructed on the basis of [27], which only supports AND gates. Suppose that the attribute universe is U={$att_1, att_2, \ldots, att_n$}, where each att_i has 2 values: "+" and "−". The "+" denotes that user owns this attribute, while the

"−" denotes that user does not own this attribute. The specific scheme is as follows:

Setup(κ, U): the setup algorithm is run by KGC. It takes security parameters κ and attribute universe U as input, where $|U| = n$. The algorithm first chooses q order bilinear groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, where g is a generator of $\mathbb{G}_1$ and h is a generator of $\mathbb{G}_2$. Let H be a cryptographic hash function; $H : \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. For $i \in [1,n]$, it chooses $r_i, r_{n+i} \in \mathbb{Z}_p^*$, $s_i, s_{n+i} \in \mathbb{G}_2$ randomly and sets $u_i = g^{-r_i}$ and $h_i = e(g, s_i)$. It outputs

$$PP := \left\{g, h, \left(u_k, h_k\right)_{k\in[1,2n]}, H\right\},$$
$$MSK := \left\{\left(r_k, s_k\right)_{k\in[1,2n]}\right\}. \tag{3}$$

In general speaking, $\{\langle u_i, h_i\rangle | i \in [1,n]\}$ correspond to the positive attributes and $\{\langle u_{n+i}, h_{n+i}\rangle | i \in [1,n]\}$ correspond to the negative attributes.

UserTemKeyGen(PP, κ): the user's temporary-key generation algorithm is run by user. It takes public parameters PP and security parameters κ as input and chooses $\beta_i \longleftarrow \mathbb{Z}_q$ randomly for $i \in [1,n]$, as its temporary secret key TSK_{User}. Then, it calculates the temporary public key $TPK_{User} = \{h^{\beta_i}\}_{i\in[1,n]}$.

BlindTokenGen(PP, S, TPK_{User}): the blind token generation algorithm is run by AAC. It takes public parameters PP, user's attributes set S, and user's temporary public key TPK_{User} as input. S expresses an attributes set, which includes n signs, e.g., $S = (+, -, +\ldots, +)$, where "+" indicates that the user owns this attribute and "−" indicates that the user does not own this attribute. It selects $\alpha_i, \gamma_i \longleftarrow \mathbb{Z}_p$ randomly and calculates h^{α_i}, $(h^{\beta_i})^{\alpha_i}$, and h^{γ_i}, for $i \in [1,n]$.

If the attribute $att_i =$"+"$\in S$, then it sets $x_i = h^{\alpha_i}$, $y_i = h^{\beta_i}$, $z_{i,0} = h^{\alpha_i\beta_i}$, $z_{i,1} = h^{\gamma_i}$. Otherwise, the attribute $att_i =$ "−" $\in S$, and it sets $x_i = h^{\alpha_i}$, $y_i = h^{\beta_i}$, $z_{i,0} = h^{\gamma_i}$, $z_{i,1} = h^{\alpha_i\beta_i}$.

$$t_S = \left\{t_i = \left(x_i \parallel y_i \parallel z_{i,0} \parallel z_{i,1}\right)\right\}_{i\in[1,n]}. \tag{4}$$

Then, it runs standard signature algorithm on t_S to get a signature Σ and returns $T_S = (t_S, \Sigma)$ to user.

BlindKenGen(PP, MSK, T_S): the user submits the token T_S corresponding to attributes set S to the KGC for applying secret key. KGC first checks that all $x_i, y_i, z_{i,0}, z_{i,1} \in \mathbb{G}_1$, $z_{i,0} \neq z_{i,1}$ and that Σ is legal. If not, it aborts outputting $\perp$; otherwise it randomly chooses $u_{i,0}, u_{i,1}, v_{i,0}, v_{i,1} \longleftarrow \mathbb{Z}_q$ for $i \in [1,n]$ and calculates the following values:

$$w_{i,0} = x_i^{u_{i,0}} \cdot h^{v_{i,0}},$$
$$k_{i,0} = \left(z_{i,0}\right)^{u_{i,0}} \cdot y_i^{v_{i,0}}$$
$$w_{i,1} = x_i^{u_{i,1}} \cdot h^{v_{i,1}}, \tag{5}$$
$$k_{i,1} = \left(z_{i,1}\right)^{u_{i,1}} \cdot y_i^{v_{i,1}}$$

Then, it randomly chooses $v \in \mathbb{G}_2$ for each user and calculates $\sigma_{i,0} = s_i v^{r_i}$, $\sigma_{i,1} = s_{n+i} v^{r_{n+i}}$, for $i \in [1,n]$. It calculates $c_{i,0} = \sigma_{i,0} \cdot k_{i,0}$, $c_{i,1} = \sigma_{i,1} \cdot k_{i,1}$, for $i \in [1,n]$.

The blind secret key

$$BSK = \left\langle v, \left\{\left(w_{i,0}, c_{i,0}\right), \left(w_{i,1}, c_{i,1}\right)\right\}_{i\in[1,n]}\right\rangle. \tag{6}$$

It returns BSK to user.

KeyExtra(TSK_{User}, BSK): the key extract algorithm is run by user.

For $i \in [1,n]$, if $att_i =$"+", it sets $w_i = w_{i,0}$, $c_i = c_{i,0}$; else if $att_i =$ " − ", it sets $w_i = w_{i,1}$, $c_i = c_{i,1}$ and calculates

$$k_i = \left(w_i\right)^{\beta_i},$$
$$\sigma_i = \frac{c_i}{k_i}. \tag{7}$$

It outputs

$$SK := \left\langle v, \left\{\sigma_i\right\}_{i\in[1,n]}\right\rangle. \tag{8}$$

We note, in the above key issuing procedure, that KGC cannot obtain the specific attributes of user, and AAC cannot obtain the secret key.

Encrypt(PK, M, $\mathbb{W}$): it takes public key PK, AND gate structure $\mathbb{W}$, and message M as input, where $\mathbb{W} = \bigwedge_{att_i\in A}\underline{att_i}$, for A is the related attributes set, and $\underline{att_i} \in \{$" + ","− "$\}$. It chooses $s \in \mathbb{Z}_p^*$ randomly and sets $\langle u_A, h_A\rangle = \langle\prod_{att_i\in A} u_{\underline{i}}, \prod_{att_i\in A} h_{\underline{i}}\rangle$, where for each $att_i \in A$,

if $att_i =$"+", $\langle u_{\underline{i}}, h_{\underline{i}}\rangle = \langle u_i, h_i\rangle$

if $att_i =$ " − ", $\langle u_{\underline{i}}, h_{\underline{i}}\rangle = \langle u_{n+i}, h_{n+i}\rangle$

Then, it computes $ct_0 = M \cdot h_A^s$, $ct_1 = g^s$, $ct_2 = u_A^s$.

The ciphertext is defined as $CT = (\mathbb{W}, ct_0, ct_1, ct_2)$.

Decrypt(PP, SK_S, CT): if $S \vDash \mathbb{W}$, the decryption algorithm computes $sk = \langle v, \sigma = \prod_{att_i\in A}\sigma_i\rangle$, for related attributes set A. Then, it computes the message

$$M = \frac{ct_0}{e\left(ct_1, \sigma\right) \cdot e\left(ct_2, v\right)}. \tag{9}$$

4.2. Correctness. The correctness is guaranteed by

$$\begin{aligned} e\left(ct_1, \sigma\right) \cdot e\left(ct_2, v\right) &= e\left(g^s, \prod_{att_i\in A}\sigma_i\right) e\left(\prod_{att_i\in A} u_{\underline{i}}^s, v\right) \\ &= \prod_{att_i\in A}\left(e\left(g, \sigma_i\right) \cdot e\left(u_{\underline{i}}, v\right)\right)^s \\ &= \prod_{att_i\in A}\left(e\left(g, s_{\underline{i}} v^{r_{\underline{i}}}\right) \cdot e\left(g^{-r_{\underline{i}}}, v\right)\right)^s \\ &= \prod_{att_i\in A}\left(e\left(g, s_{\underline{i}}\right) \cdot e\left(g, v^{r_{\underline{i}}}\right) \cdot e\left(g^{-r_{\underline{i}}}, v\right)\right)^s \\ &= \prod_{att_i\in A}\left(e\left(g, s_{\underline{i}}\right)\right)^s = \prod_{att_i\in A} h_{\underline{i}}^s = h_A^s. \end{aligned} \tag{10}$$

5. Proof of Security

5.1. Confidentiality

Theorem 8. *If the decisional n−BDHE assumption holds for bilinear groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, our PPGK-ABE scheme is selectively IND-CPA secure.*

Proof. If the adversary $\mathcal{A}$ can win above security game with nonnegligible advantage, we can construct an algorithm $\mathcal{B}$ to break the decision n−BDHE assumption. $\mathcal{B}$ plays the security game with $\mathcal{A}$ as follows:

Init. $\mathcal{B}$ receives challenge gate $\mathbb{W}^* = \bigwedge_{att_{i_j} \in A^*} \overline{att_{i_j}}$ from $\mathcal{A}$. We suppose $|A^*| = m < n$, and $I^* = \{i_j | att_{i_j} \in A^*\}$.

Setup. $\mathcal{B}$ chooses $j^* \in [1, m]$, $r'_k \in \mathbb{Z}_p^*$, $x_k \in \mathbb{G}_2$, $r_{i_j}, a_{i_j} \in \mathbb{Z}_p^*$ randomly, for $k \in [1, 2n]$, $j \in [1, m]$.

For $i_j \in I^* - \{j^*\}$, $\mathcal{B}$ computes public parameters as follows:

(1) If $\underline{att_{i_j}} = +$,

$$\begin{aligned} \left(u_{i_j}, h_{i_j}\right) &= \left(g^{r_{i_j}} g_{n+1-i_j}^{-1}, e(g,h)^{a_{i_j}}\right), \\ \left(u_{i_j+n}, h_{i_j+n}\right) &= \left(g^{-r'_{i_j+n}}, e\left(g, x_{i_j+n}\right)\right). \end{aligned} \tag{11}$$

(2) If $\underline{att_{i_j}} = -$,

$$\begin{aligned} \left(u_{i_j}, h_{i_j}\right) &= \left(g^{-r'_{i_j}}, e\left(g, x_{i_j}\right)\right), \\ \left(u_{i_j+n}, h_{i_j+n}\right) &= \left(g^{r_{i_j}} g_{n+1-i_j}^{-1}, e(g,h)^{a_{i_j}}\right). \end{aligned} \tag{12}$$

For $att_{i_j} = att_{i_{j^*}}$, $\mathcal{B}$ computes as follows:

(1) If $\underline{att_{i_{j^*}}} = +$,

$$\begin{aligned} &\left(u_{i_{j^*}}, h_{i_{j^*}}\right) \\ &= \left(g^{r_{i_{j^*}}} \cdot \prod_{k \in I^* - \{i_{j^*}\}} g_{n+1-k}, e(g,h)^{a_{i_{j^*}}} e(g,h)^{\alpha^{n+1}}\right) \\ &\left(u_{i_{j^*+n}}, h_{i_{j^*+n}}\right) = \left(g^{-r'_{i_{j^*}+n}}, e\left(g, x_{i_{j^*}+n}\right)\right). \end{aligned} \tag{13}$$

(2) If $\underline{att_{j^*}} = -$,

$$\begin{aligned} &\left(u_{i_{j^*}}, h_{i_{j^*}}\right) = \left(g^{-r'_{i_{j^*}}}, e\left(g, x_{i_{j^*}}\right)\right), \\ &\left(u_{i_{j^*+n}}, h_{i_{j^*+n}}\right) \\ &= \left(g^{r_{i_{j^*}}} \cdot \prod_{k \in I^* - \{i_{j^*}\}} g_{n+1-k}, e(g,h)^{a_{i_{j^*}}} e(g,h)^{\alpha^{n+1}}\right), \end{aligned} \tag{14}$$

For $i_j \notin I^*$, $\mathcal{B}$ computes

$$\begin{aligned} \left(u_{i_j}, h_{i_j}\right) &= \left(g^{-r'_{i_j}}, e\left(g, x_{i_j}\right)\right), \\ \left(u_{i_j+n}, h_{i_j+n}\right) &= \left(g^{-r'_{i_j+n}}, e\left(g, x_{i_j+n}\right)\right). \end{aligned} \tag{15}$$

□

Phase 9. $\mathcal{A}$ queries secret key on any attributes set $S \not\models \mathbb{W}^* = \bigwedge_{att_{i_j} \in A^*} \overline{att_{i_j}}$; therefore, $\exists att_{i_j} \in \mathrm{A}^*$, s.t. either $att_{i_j} \in S$ for $\underline{att_{i_j}} = -$, or $att_{i_j} \notin S$ for $\underline{att_{i_j}} = +$. Suppose that $\overline{att_{i_j}} \in S \not\models \mathbb{W}$. $\mathcal{B}$ chooses $z \in \mathbb{Z}_p^*$ randomly and computes $\nu = g_{i_j} g^z$.

For att_{i_j}, $\mathcal{B}$ computes $\sigma_{att_{i_j}} = x_{i_j+n} (g_{i_j} g^z)^{r'_{i_j+n}}$.

For $att_{\mathrm{i}} \neq att_{i_j}$, σ_{att_i} is computed as follows:

(1) If $i = i_k \in I^* - \{j^*\}$ $(k \neq j^*)$, calculate

$$\sigma_{att_i} = g^{a_{i_k}} \left(g_{i_j}\right)^{r_{i_k}} g_{n+1-i_k+i_j} \left(u_{i_k}\right)^{-z}. \tag{16}$$

(2) If $i = i_{j^*}$, calculate

$$\sigma_{att_i} = g^{a_{i_{j^*}}} \left(g_{i_j}\right)^{r_{i_{j^*}}} \left(\prod_{k \in I^* - \{j^*\}}^{k \neq i_j}\right) \left(u_{i_j^*}\right)^{-z}. \tag{17}$$

(3) If $i \notin I^*$, calculate

(a) $\sigma_{att_i} = x_i (g_{i_j} g^z)^{r'_i}$, if $\underline{att_i} = +$;

(b) $\sigma_{att_i} = x_{i+n} (g_{i_j} g^z)^{r'_{i+n}}$, if $\underline{att_i} = -$.

$\mathcal{B}$ answers secret key query for S:

$$SK = \left\langle \nu, \left\{\sigma_{att_i} \mid i \in [1, n]\right\}\right\rangle. \tag{18}$$

Challenge. For $a_{I^*} = \sum_{j=1}^{m} a_{i_j}$, $r_{I^*} = \sum_{j=1}^{m} r_{i_j}$, $\langle u_{I^*}, h_{I^*} \rangle$ is calculated as follows:

$$\begin{aligned} u_{I^*} &= \underline{u_{i_{j^*}}} \prod_{k \in I^* - \{j^*\}} \underline{u_k} \\ &= \left(g^{r_i j^*} \prod_{k \in I^* - \{j^*\}} g_{n+1-k}\right) \prod_{k \in I^* - \{j^*\}} g^{r_k} g_{n+1-k}^{-1} = g^{r_{I^*}}, \\ h_{I^*} &= \underline{h_{i_{j^*}}} \prod_{k \in I^* - \{j^*\}} \underline{h_k} \\ &= e(g,h)^{a_{i_{j^*}}} \cdot e(g,h)^{\alpha^{n+1}} \prod_{k \in I^* - \{j^*\}} e(g,h)^{a_k} \\ &= e(g,h)^{a_{I^*} + \alpha^{n+1}}. \end{aligned} \tag{19}$$

$\mathcal{A}$ submits M_0 and M_1 with $|M_0| = |M_1|$. $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$, $s' \in \mathbb{Z}_p^*$ and computes

$$\begin{aligned} CT^* &= \left(\mathbb{W}^*, ct_0^* = M_b T e(g,h)^{s' a_{I^*}}, ct_1^* = g^{s'}, ct_2^*\right. \\ &\quad \left. = g^{s' r_{I^*}}\right). \end{aligned} \tag{20}$$

If $T = e(g_{n+1}, h)$, CT^* is a valid ciphertext; else if T is random, CT^* is independent of b.

Guess. If $\mathcal{A}$ outputs $b' = b$, $\mathcal{B}$ guesses that $T = e(g_{n+1}, h)$. Otherwise, $\mathcal{B}$ guesses that T is random.

Therefore, $\mathcal{B}$ can break the decisional n−BDHE assumption with nonnegligible advantage.

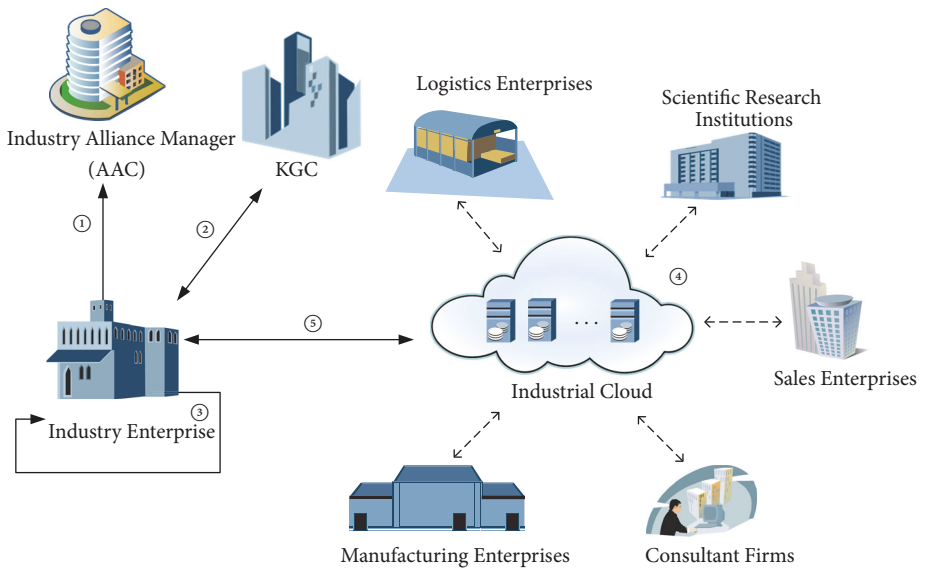


Figure 2: Application in industrial cloud.

5.2. Privacy

Theorem 10. *If the DDH assumption holds in $\mathbb{G}_2$, our PPGK-ABE scheme is privacy preserving in key generation phase.*

Proof. If DDH assumption holds in $\mathbb{G}_2$, no probabilistic polynomial-time adversary can distinguish following tuple: $(h^{\alpha}, h^{\beta}, h^{\alpha\beta}, h^{\gamma})$ and $(h^{\alpha}, h^{\beta}, h^{\gamma}, h^{\alpha\beta})$, where h is a generator of group $\mathbb{G}_2$, and α, β, γ are selected from Z_p^* randomly. Therefore, no probabilistic polynomial-time adversary can win the security game for privacy. □

6. Application in Industrial Cloud

Nowadays, new technological revolution represented by Big Data, cloud computing, and Internet of Things is changing the traditional industrial manufacturing system [28, 29]. Industrial cloud provides more convenient and secure cooperation model for the industrial enterprises [30–32]. The ABE scheme has been gradually used in the industrial cloud environment. In these applications, the qualifications, patents, and procurement plans owned by an enterprise often represent its attributes. Using traditional ABE system, the enterprise has to disclose these attributes' information that may relate to business secret to the KGC for applying the corresponding private key. Our PPGK-ABE scheme can solve this problem correctly. In this section, we introduce how to deploy our scheme in the industrial cloud environment. Figure 2 shows the specific structure of the application using our PPGK-ABE scheme. It consists of the following entities:

(i) Industry Enterprise: in this system, the role of industry enterprise is data user. They want to get useful information according to their business, but they do not want to reveal their attributes information that may relate to their business secret to KGC.

(ii) Industry Alliance Manager: in this system, the role of the industry alliance manager is AAC, which issues blind token for the attributes of industry enterprises after reviewing the relevant evidence.

(iii) KGC: its responsibility is to issue the corresponding key to the attributes of industry enterprises. In this system, KGC cannot get these attributes.

(iv) Industrial Information Provider: in this system, industrial information providers are the members of the industrial alliance and include manufacturing enterprises, sales enterprises, logistics enterprises, scientific research institutions, consultant firms, and so on. They will use ABE scheme to share their encrypted data.

(v) Industrial Cloud: industrial cloud serves as data storage center and data sharing center in this system. In order to protect security and privacy of industrial Big Data, the industrial information providers upload their data in encrypted form.

The specific workflow is as follows:

(1) After checking the relevant evidence, the industry enterprise and the industry alliance manager run *UserTemKeyGen* and *BlindTokenGen* algorithms, respectively. The industry enterprise gets the blind token corresponding to its attributes.

(2) When the industry enterprises need to ask for their attributes keys, they will submit their blind tokens to KGC. The KGC runs *BlindKenGen* algorithm and returns blind secret keys to the industry enterprises. In this process, the KGC cannot get any information about the enterprises' attributes.

(3) After receiving the blind secret keys, the industry enterprises run *KeyExtra* algorithm to obtain their

own secret key. Even if the industry alliance manager knows the attributes of industry enterprises, it does not know the secret keys corresponding to these attributes.

(4) The industrial information providers run *Encrypt* algorithm to encrypt the industrial data based on some access policies. Then, they share encrypted data on the cloud. Only the enterprises that meet the policies can access corresponding data.

(5) The industry enterprises acquire encrypted data from the cloud and run *Decrypt* algorithm to get plaintext.

In the above application, industrial information providers can share industrial data according to enterprises' attributes. Only the enterprises that meet the access policy are able to access data. Unlike traditional ABE solutions, in this application, the attributes information of enterprises will not be known by KGC. The business secret of enterprises is protected.

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 they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61602287, No. 61802235, No. 61672330, and No. 61702168), the Primary Research & Development Plan of Shandong Province (No. 2018GGX101037), and the Major Scientific and Technological Innovation Project of Shandong Province (No. 2018CXGC0702).

References

[1] F. Wang, J. Mickens, N. Zeldovich, and V. Vaikuntanathan, "Sieve: Cryptographically enforced access control for user data in untrusted clouds," in *Proceedings of the13th USENIX Symposium on Networked Systems Design and Implementation, (NSDI, '16)*, pp. 611–626, Santa Clara, Calif, USA, 2016.

[2] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating eves algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[3] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '05*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Aarhus, Denmark, 2005.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, (CCS '06)*, pp. 89–98, Alexandria, VA, USA, November 2006.

[6] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '10*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91, Springer, Monaco, French Riviera, 2010.

[7] H. Wang, Z. Zheng, L. Wu, and D. He, "New large-universe multi-authority ciphertext-policy ABE scheme and its application in cloud storage systems," *Journal of High Speed Networks*, vol. 22, no. 2, pp. 153–167, 2016.

[8] H. Wang, D. He, J. Shen, Z. Zheng, C. Zhao, and M. Zhao, "Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing," *Soft Computing*, vol. 21, no. 24, pp. 7325–7335, 2017.

[9] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the CRYPTO '84 Advances in Cryptology*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Santa Barbara, Calif, USA, 1984.

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '07)*, pp. 321–334, Oakland , Calif, USA, 2007.

[11] H. Wang, D. He, J. Shen, Z. Zheng, X. Yang, and M. H. Au, "Fuzzy matching and direct revocation: a new CP-ABE scheme from multilinear maps," *Soft Computing*, vol. 22, no. 7, pp. 2267–2274, 2018.

[12] H. Wang and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 42, no. 8, pp. 152:1–152:9, 2018.

[13] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[14] D. J. Lehmann and M. O. Rabin, "On the advantages of free choice: A symmetric and fully distributed solution to the dining philosophers problem," in *Proceedings of the Conference Record of the Eighth Annual ACM Symposium on Principles of Programming Languages*, pp. 133–138, Williamsburg, Va, USA, January 1981.

[15] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[16] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proceedings of the CRYPTO '86 - Advances in Cryptology*, vol. 263 of *Lecture Notes in Comput. Sci.*, pp. 234–238, Springer, Santa Barbara, Calif, USA, 1986.

[17] J. P. Stern, "A new efficient all-or-nothing disclosure of secrets protocol," in *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security, Advances in Cryptology - ASIACRYPT '98*, vol. 1514 of *Lecture Notes in Computer Science*, pp. 357–371, Springer, Beijing, China, 1998.

[18] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms*, pp. 448–457, SIAM, Washington, DC, USA, 2001.

[19] B. Aiello, Y. Ishai, and O. Reingold, "Priced oblivious transfer: How to sell digital goods," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT '01*, vol. 2045 of *Lecture Notes in Comput. Sci.*, pp. 119–135, Springer, Innsbruck, Austria, 2001.

[20] W.-G. Tzeng, "Efficient 1-out-of-n oblivious transfer schemes with universally usable parameters," *IEEE Transactions on Computers*, vol. 53, no. 2, pp. 232–240, 2004.

[21] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Proceedings of the 23rd Annual International Cryptology Conference-Advances in Cryptology - CRYPTO '03*, vol. 2729, pp. 145–161, Springer, Santa Barbara, Calif, USA, 2003.

[22] H. Qin, H. Wang, X. Wei, L. Xue, and L. Wu, "Privacy-preserving wildcards pattern matching protocol for IoT applications," *IEEE Access*, vol. 7, pp. 36094–36102, 2019.

[23] Q. Wang, L. Gao, H. Wang, and X. Wei, "Face detection for privacy protected images," *IEEE Access*, vol. 7, pp. 3918–3927, 2019.

[24] H. Xia, J. Yu, C.-L. Tian, Z.-K. Pan, and E. Sha, "Light-weight trust-enhanced on-demand multi-path routing in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 62, pp. 112–127, 2016.

[25] H. Xia, J. Yu, Z.-K. Pan, X.-G. Cheng, and E. H.-M. Sha, "Applying trust enhancements to reactive routing protocols in mobile ad hoc networks," *Wireless Networks*, vol. 22, no. 7, pp. 2239–2257, 2016.

[26] C. Hazay and Y. Lindell, *Efficient Secure Two-Party Protocols*, Information Security and Cryptography, Springer, Berlin, Germany, 2010.

[27] C. Chen, Z. Zhang, and D. Feng, "Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost," in *Proceedings of the 5th International Conferenceo on Provable Security, ProvSec '11*, vol. 6980 of *Lecture Notes in Computer Science*, pp. 84–101, Springer, Xi'an, China, 2011.

[28] H. Xia, S. Zhang, B. Li, L. Li, and X. Cheng, "Towards a novel trust-based multicast routing for VANETs," *Security and Communication Networks*, vol. 2018, Article ID 7608198, 12 pages, 2018.

[29] H. Xia, C. Hu, F. Xiao, X. Cheng, and Z. Pan, "An efficient social-like semantic-aware service discovery mechanism for large-scale Internet of Things," *Computer Networks*, vol. 152, pp. 210–220, 2019.

[30] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. P. C. Rodrigues, "SDN-Enabled multi-attribute-based secure communication for smart grid in IIoT environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2629–2640, 2018.

[31] M. Zhang, Y. Yao, Y. Jiang, B. Li, and C. Tang, "Accountable mobile E-commerce scheme in intelligent cloud system transactions," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1889–1899, 2018.

[32] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 9126376, 12 pages
https://doi.org/10.1155/2019/9126376

Research Article

Lightweight Privacy Preservation for Securing Large-Scale Database-Driven Cognitive Radio Networks with Location Verification

Rui Zhu,[1] **Li Xu**,[1] **Yali Zeng**,[1] **and Xun Yi**[2]

[1] *Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, China*
[2] *The School of Science, RMIT University, Australia*

Correspondence should be addressed to Li Xu; xuli@fjnu.edu.cn

Received 21 February 2019; Accepted 17 April 2019; Published 6 May 2019

Guest Editor: Mingwu Zhang

The database-driven cognitive radio networks (CRNs) are regarded as a promising approach to utilizing limited spectrum resources in large-scale Internet of Things (IoT). However, database-driven CRNs face some security and privacy threats. Firstly, secondary users (SUs) should send identity and location information to the database (DB) to obtain a list of available channels, such that the curious DB might easily misuse and threaten the privacy of SUs. Secondly, malicious SUs might send fake location information to the DB in order to occupy channels with better quantity in advance and so gain benefits. This might also cause serious interference to primary users (PUs). In this paper, we propose a lightweight privacy-preserving location verification protocol to protect the identity and location privacy of each SU and to verify the location of SUs. In the proposed protocol, the SU does not need to provide location information to request an available channel from the DB. Therefore, the DB cannot get the location information of any SU. In the proposed protocol, the base station (BS) selects some SUs as witnesses to generate location proofs for each other in a distributed fashion. This new witness selection mechanism makes the proposed protocol reliable when a malicious SU generates fake location information to cheat the BS and also prevents SU-Witness collusion attacks. The results also show that the proposed protocol can provide strong privacy preservation for SUs and can effectively verify the location of the SUs. The security analysis shows that the proposed protocol can resist various types of attacks. Moreover, compared with previous protocols, the proposed protocol is lightweight because it relies on symmetric cryptography and it is unaffected by the area covered by the DB.

1. Introduction

The explosive growth of wireless devices and services has exacerbated the spectrum scarcity problem in large-scale Internet of Things (IoT) [1]. Database-driven CRNs are regarded as a promising approach that allows the dynamic spectrum sharing in many large-scale IoT applications [2]. In cognitive radio networks (CRNs), primary users (PUs) have exclusive privilege to access the licensed channels, while secondary users (SUs) are allowed to access the licensed channels when the PUs are off-line or at a lower power without causing interference to the PUs. Through spectrum sharing between PUs and SUs, CRNs can effectively improve the spectrum utilization and alleviate the spectrum scarcity crisis.

CRNs have many potential applications including IoT [3], smart cities [4], and vehicular networks [5]. For example, a large number of connected devices will create a major challenge in terms of spectrum scarcity in large-scale IoT. Through dynamic spectrum access capabilities, large-scale IoT not only improves spectrum utilization but also exploits alternate spectrum opportunities. In addition, cognitive IoT is inherently equipped to address the challenges of interference management, energy efficiency, and device heterogeneity.

In CRNs, two main approaches can be used by SUs to obtain channel availability information [6]: the spectrum sensing approach and the geolocation database-driven approach. In the sensing approach [7], several SUs cooperatively sense the idle channels and report these channels to the

fusion center; the latter will then decide whether a channel is available prior to its use so as to avoid interfering with PUs. In the database-driven approach [7, 8], the database (DB) that is administered by some commercial entities (e.g., Microsoft, Google Inc., and Cellular South) stores the spectrum available information (SAI). SUs are required to send their locations to the DB in order to obtain channel availability information. After receiving a query, the DB returns a list of available channels and some transmission parameters to the SU at its location.

In 2012, database-driven CRNs were specified as the primary approach by the Federal Communications Commission (FCC) [8] because the geolocation database-driven approach has more advantages, especially in large-scale CRNs. On the one hand, it pushes the responsibility and complexity spectrum policies conformity from SUs to DBs. On the other hand, it only updates a handful of DB when policy changes, instead of updating large number of devices [9]. However, SUs have to send location information to the DB to obtain the list of available channels. Through this location information, the DB might reveal SUs' critical information such as habits, shopping preferences, behavior, and commuter routes [10]. Therefore, it is urgent to protect the location privacy of the SU during the spectrum query in CRNs. Some protocols based on private information retrieval (PIR) [11, 12] or k-anonymity [13, 14] have been proposed to protect the location privacy of SUs. However, these protocols might cause high computation or communication overheads or cannot provide enough safeguards for location privacy of SUs.

On the other hand, malicious SUs may falsify their locations when querying the DB for the list of available channels [15]. Because there is no location proof and verification when SUs query channels in database-driven CRNs, a malicious SU might send fake location information to the DB in order to occupy channels with better quantity in advance and gain benefits. Besides, if a malicious SU chooses channels that are not in his/her real location, it may cause interference to PUs. Only a few protocols based on private information retrieval [16] or cryptography [2] focus on the location verification: most existing protocols achieve location verification by employing a large number of access points (APs) to be witnesses. It might be too expensive and be inapplicable in some scenarios where no fixed wireless infrastructures are accessible [17]. Moreover, these protocols do not consider collusion between SUs and witnesses.

Noting all these concerns, we propose a lightweight privacy-preserving location verification protocol (LPPLV) for large-scale database-driven CRNs. The proposed protocol aims to guarantee the privacy of SUs to avoid inference by PUs during spectrum queries in database-driven CRNs. To prevent malicious SUs from reporting fake location information, SUs are required to provide location proofs for the DB indicating that they are at the places where they claim to be.

The contributions of this paper are summarized as follows.

(1) We propose a secure and privacy-aware protocol that can protect the identity and location privacy of SUs. In our protocol, the DB has no knowledge about the location information of SUs, and this can prevent the DB from obtaining the real identity and location information of SUs.

(2) Our proposed protocol has a distributed architecture in which SUs generate location proofs for each other instead of employing expensive APs to provide the location proof for SUs in existing protocols. The new witness selection mechanism allows malicious SUs to be discovered by the BS when malicious SUs generate fake location information. Our new witness selection mechanism also reduces the possibility of SU-Witness collusion.

(3) Security analysis is presented to prove that our protocol resists various types of attacks in database-driven CRNs, such as replaying attack, Man-in-the-Middle attack, and eavesdropping attack.

(4) Channel allocation is implemented locally in our protocol, and we use the symmetric encryption operation in the communication of database-driven CRNs. Therefore, our protocol improves the efficiency of obtaining available channels for SUs. Experiments' results indicate the efficiency of our privacy-preserving location proof protocol.

The rest of this paper is organized as follows. We first discuss related work in Section 2. Section 3 presents the preliminaries. Then we introduce our proposed protocol in Section 4, which is followed by security and privacy analysis in Section 5. Section 6 analyzes the performance of the proposed protocol. Finally, Section 7 concludes the paper.

2. Related Work

Most of the existing protocols that are related to this work focus on privacy preservation of SUs while some protocols focus on location verification of SUs. In this section, we give a brief overview of these protocols.

2.1. Privacy Preservation Protocols. PIR [18], k-anonymity [19], and cuckoo filter [11] are three techniques that can be used to protect the privacy of SUs. Gao et al. [20] proposed a PIR-based protocol, in which SUs can obtain available spectrum information from the DB without sending geographic location information. They also pointed out that if the SUs switch channels frequently, the DB can locate the SUs according to the channel requesting messages, which can cause user privacy leakage. Troja et al. [12] proposed another PIR-based protocol that allows SUs to share information in a peer-to-peer manner. PIR technology can protect the privacy of SUs' locations well, but it also causes high communication and computational overheads. Grissa et al. [21] proposed a cuckoo filter-based protocol: the DB uses a cuckoo filter to compress the spectrum information and send it to the query server. The SUs then obtain the available channels through the query server. However, in this protocol, SUs query efficiency and spectrum accuracy are affected by cuckoo filters false positive and negative rates. Petrov et al. [13] proposed a privacy protection protocol based on k-anonymity. In this

protocol, SUs select k-1 volunteers to form a link forwarding query message, so that the DB cannot distinguish the true identity of the user who sent the query, achieving the purpose of privacy protection. However, this method creates high communication overheads, and when the number of volunteers is too small, SUs can easily expose their true identities and cannot achieve good privacy protection.

2.2. Location Verification Protocols. Another focus of this paper is the problem of location verification of SUs in database-driven CRNs. Numerous protocols have been proposed to achieve the location verification [17, 22–25]. SUs need to provide location proofs before obtaining the available channel in order to secure the communication in database-driven CRNs. Xin et al. [16] provide a PIR-based protocol for the server to verify whether the query SU is located where it claims to be. It uses WiFi access points (APs) to provide location proofs. In particular, if an AP can communicate directly with an SU, it considers this SU to be at the same location as itself. Then it generates a signature as the location proof for this SU. Li et al. [2] proposed a protocol based on the public key cryptographic algorithm. That protocol also relies on APs to provide location proofs. The two protocols have high computation and communication overheads. Moreover, it might be too expensive and is inapplicable to some scenarios in which no fixed wireless infrastructure is accessible.

In our previous work, we mainly focus on the privacy preservation and authentication of SUs. Zeng et al. [9, 26], respectively, utilize elliptic curve cryptography (ECC) and modular square root techniques to achieve privacy preservation and authentication of SUs in database-driven CRN. However, we did not take into account the problem of location verification of SUs.

From the above discussion, we can see that these protocols either incur high communication and computation overheads or rely on existing WiFi AP networks for reporting location-based activity summaries, which means that these existing protocols cannot provide a satisfactory privacy preservation and location proof protocol for SUs in database-driven CRNs.

3. Preliminaries

3.1. System Model. Our network model is shown in Figure 1, which involves four types of entities: the database (DB), multiple based stations (BSs), numerous secondary users (SUs), and the trust authority (TA). Their functions and interactions are described as follows.

DB. The DB is an entity which stores the spectrum available information and makes this information available to SUs.

BSs. In our network model, BSs provide location verification and channel allocation in our system. Each BS requests some channels from the DB. If an SU wants to obtain available channels, it should send a request including its location information to the BS. The BS first verifies its location proof, i.e., this SU is where it claims to be. Then, the BS calculates and allocates a channel based on the SU's location.

SUs. The SUs are those who do not have a licensed spectrum in CRNs. There are three types of SUs: one who wants to obtain an available channel, one who is selected to be a witness to provide location proofs for the SU, and one who is nonselected to be witness. Multiple SUs will be selected as witnesses after an SU sends a request to the BS. Meanwhile, that SU does not want to disclose its location. Besides, SUs and witnesses send their messages to each other through their predefined short-range communication interfaces (e.g., Bluetooth and WiFi).

TA. The TA can be served by federal technical centers (e.g., FCC and NTIA). In this paper, each BS and SU are required to be registered in the TA. The TA stores the reputation of each SU and updates it for a period of time.

A typical process of our protocol can be described as follows. BSs first obtain the list of available channels from the DB. An SU should ask the BS to obtain a channel with its current location. Then the BS chooses some SUs who are using the channels and are near the SU as witnesses. After obtaining the location proofs for the SU from witnesses, the BS verifies the location proof. If it is valid, the BS chooses a channel and returns the channel to the SU.

3.2. Security and Privacy Requirements. The TA in our system is a fully trusted entity which has the knowledge of the mappings between the pseudo-ID and the real identity of each SU. We also assume that BSs are supposed to be trusted entities. Neither the TA nor the BS publishes SUs' identity and data. However, the DB is honest but curious, which means the DB will honestly execute the protocol but is very curious about the identity and location information of the SUs. Hence, our first goal is to prevent the DB from obtaining any SU's identity and location information. Meanwhile, malicious SU might provide witnesses with fake information about his/her location or change contents of a location proof generated for him/her or another SU. They might also collude with witnesses to cheat the BS and obtain the available channels. It is also assumed that both the DB and BSs will not modify the communication data between them and legitimate SUs. No entity ever shares its private keys with another.

An efficient privacy-preserving location verification protocol for the large-scale database-driven CRNs should have the following desirable properties.

Privacy preservation: The privacy we want to preserve is the identity and location information of SUs. That is, except the TA, other entities including the DB, the BSs, and other SUs will never get the knowledge of the SU's real identity. On the other hand, our protocol should prevent the DB from obtaining the location information from the channel query of the SUs, even if the DB colludes with all BSs and other SUs.

Location verification: Malicious SUs might send fake location information to the DB and obtain the channels that are not available in their real locations, so as to infer operational patterns of PUs and other SUs. Therefore, the proposed protocol should verify the SUs' current locations and prevent the SUs from obtaining any available channels at other locations.

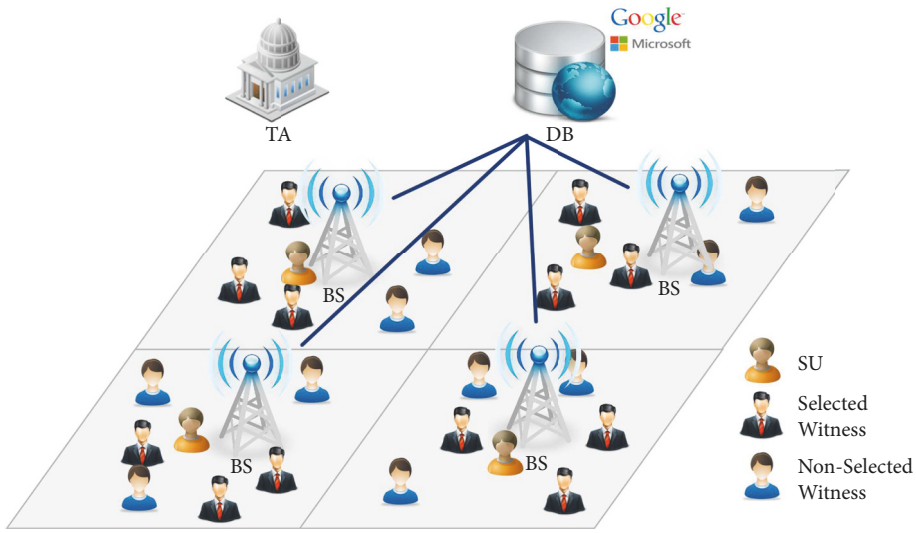


Figure 1: Network model.

Attack resistance: There are various types of attacks in CRNs, such as replaying attack, Man-in-the-Middle attack, and eavesdropping attack. The security of our protocol will not be compromised under these attacks.

SU-Witness collusion: Malicious SU might collude with malicious witnesses by creating a location proof for him/her even though the SU is not at the location claimed, in order to deceive the BS and the DB. We call this collusion SU-Witness collusion. To the best of our knowledge, there is no protocol to consider SU-Witness collusion in CRNs.

4. The Proposed Protocol

In this section we describe the details of the proposed protocol. It consists of seven phases: system initialization, channel preallocation, channel query, location proof generation, location verification, channel allocation, and reputation update. Notations used in the protocol are defined in Table 1.

4.1. System Initialization Phase. We assume that a network is composed of N_1 BSs and N_2 SUs. Each BS and SU send their real identities to the TA in a secure channel for registration.

For each BS_i ($i \in N_1$) whose real identity is ID_{BS_i}, the TA generates the secret key SK_{BS_i} and public key PK_{BS_i}. The TA also generates a unique session key k_{DB,BS_i} between the DB and the BS_i. Then TA sends SK_{BS_i} and k_{DB,BS_i} to BS_i in a secure channel and publishes PK_{BS_i}. The TA sends k_{DB,BS_i} to the DB. Upon receiving SK_{BS_i} and k_{DB,BS_i}, BS_i stores these keys securely.

For each SU_j ($j \in N_2$) whose real identity is ID_{SU_j}, the TA chooses X distinct pseudo-IDs (PIDs) $PID_{SU_{j,1}}$, $PID_{SU_{j,2}}, \ldots, PID_{SU_{j,X}}$ and sends them to SU_j. The TA initializes the reputation value $Rep_{SU_j} = 0$. Upon receiving PIDs, SU_j stores the PIDs securely and uses one PID for each request. After all PIDs are allocated, SU_j asks for another X PIDs from the TA.

At the outset of a new system, there are not enough SUs in the network that can be selected as witnesses. At this time, the BS presets some trusted fixed devices (e.g., the roadside units and cellular tower) distributed in its coverage area. However, it might be too expensive to employ the trusted fixed devices all the time. We will reduce the employed fixed devices when the number of SUs in the network increases to a more practical number.

4.2. Channel Preallocation Phase. After registration, BS_i sends a request to the DB for enabling channel preallocation. This phase includes the following steps.

(1) BS_i sends the following message m_1 to the DB for enabling channel preallocation.

$$m_1 = ID_{BS_i} \| t_1 \| E_{k_{DB,BS_i}}\left(ID_{BS_i} \| Loc_{BS_i} \| t_1\right) \| MAC_{k_{DB,BS_i}}\left(m_1'\right), \tag{1}$$

where $m_1' = (ID_{BS_i} \| Loc_{BS_i} \| t_1)$, Loc_{BS_i} is the location of BS_i, and $MAC_{k_{DB,BS_i}}(m_1')$ is message authentication code of m_1' using the key k_{DB,BS_i}.

(2) After receiving the message from BS_i, the DB decrypts the message with the key k_{DB,BS_i} that matches BS_i and checks whether t_1 is a valid time stamp. Then the DB computes $MAC_{k_{DB,BS_i}}(m_1')$ and verifies that it is the same as $MAC_{k_{DB,BS_i}}(m_1')$ received through m_1. If one or both of the verifications do not hold, the DB drops this message. Otherwise, BS_i is authenticated by the DB. The DB chooses a channel list Ch_{BS_i} based on Loc_{BS_i} and sends the message m_2 to the BS_i using a secure channel.

$$m_2 = ID_{BS_i} \| t_2 \| E_{k_{DB,BS_i}}\left(Ch_{BS_i} \| t_2\right) \| MAC_{k_{DB,BS_i}}\left(m_2'\right), \tag{2}$$

where $m_2' = (Ch_{BS_i} \| t_2)$.

Table 1: Notations in the protocol.

Notation	Description
$\|$	Concatenation symbol
ID_U	Identity of entity U
PID_U	The pseudo-ID of entity U
N_1	The number of BSs
N_2	The number of SUs
PK_U	The public key of entity U
SK_U	The secret key of entity U
$E_{K_U}(m)$	Encryption of message m using key of entity U
$MAC_{K_U}(m)$	Message authentication code of message m using key of entity U
Rep_U	The reputation value of entity U
Loc_U	The location of entity U
t_i	The time stamp
CL_i	A channel list
Ch_i	An available channel
k_{U_1,U_2}	The session key between entity U_1 and U_2
r_i	A random number
$H()$	A secure hash function

(3) Upon receiving the message m_2 from the DB, BS_i checks whether the ID_{BS_i} is its identity. Then BS_i uses its secret key SK_{BS_i} to decrypt the message m_2 and verifies $MAC_{k_{DB,BS_i}}(m_2')$. If these verifications do not hold, BS_i drops this message. Otherwise, BS_i obtains the channel list CL_{BS_i}. Then the BS_i stores the channel list securely.

4.3. Channel Query Phase. Before an SU accesses the channel, the SU sends a request to the BS and informs the BS that he/she wants to get the location verification. This phase includes the following steps.

(1) SU_j randomly selects a PID from $\{PID_{SU_{j,1}}, PID_{SU_{j,2}}, \ldots, PID_{SU_{j,X}}\}$, generates a unique session key k_{SU_j,BS_i} between himself/herself and the BS_i, and sends a location verification request m_3 to BS_i.

$$m_3 = E_{PK_{BS_i}}\left(PID_{SU_j} \| Loc_{SU_j} \| k_{SU_j,BS_i} \| t_3\right) \| MAC_{k_{SU_j,BS_i}}\left(m_3'\right), \tag{3}$$

where $m_3' = (PID_{SU_j} \| Loc_{SU_j} \| k_{SU_j,BS_i} \| t_3)$.

(2) Upon receiving the message m_3 from SU_j, BS_i uses its secret key SK_{BS_i} to decrypt the message m_3. BS first checks whether the location of SU_j has the available channels. If not, BS_i responds with the following message m_4 to reject the request. Otherwise, BS_i executes the location proof generation steps described in Section 4.4.

$$m_4 = E_{k_{SU_j,BS_i}}\left(PID_{SU_j} \| Loc_{SU_j} \| t_4 \| null\right) \| MAC_{k_{SU_j,BS_i}}\left(m_4'\right), \tag{4}$$

where $m_4' = (PID_{SU_j} \| Loc_{SU_j} \| t_4 \| null)$.

4.4. Location Proof Generation Phase. Since each SU has to request location verification and available channels from BS_i, BS_i stores PIDs, locations of the SUs who have requested the channels, and the unique session keys k_{W_j,BS_i} between these SUs and BS_i. k_{W_j,BS_i} is the key generated by these SUs in their channel query phase, which is similar to k_{SU_j,BS_i}. This phase includes the following steps.

(1) Upon receiving the message m_3, BS_i searches for K SUs who are using the channel and are near SU_j. Then, BS_i sends a list of the PIDs of these SUs and PID_{SU_j} to the TA. After receiving the list and PID_{SU_j} from BS_i, the TA first searches for the real identity of SU_j and verifies whether SU_j has registered with it. Then the TA responds with the reputation values of these SUs to BS_i according to the PIDs. If SU_j has not registered in the TA, the TA rejects BS_i.

(2) We assume Rep_T is the threshold of reputation value. If $Rep_{SU_j} < Rep_T$, that means SU_j has a high probability of being dishonest. BS_i sorts K SUs according to their reputation values and checks the top N $(N \le K)$ SUs' reputation values. We assume N_H is the threshold number of location proofs that BS_i will receive. If the number of dishonest SUs is more than threshold N_H, BS_i will remove some SUs who have lower reputation values. BS_i then chooses some preset trusted fixed devices as a supplement to ensure that the number of dishonest selected witnesses is less than N_H. At the beginning of the system, there are not enough SUs in the network that can be selected as witnesses. At this time, BS_i selects some preset trusted fixed devices as witnesses.

After selecting N witnesses, BS_i generates a unique index (IND_P) for this location proof, numbers selected witnesses to n_i, and generates the session keys $k_{W_k,SU_j}(k \in N)$ for his/her communication process. Then BS_i sends the following message m_5 to SU_j and sends the message m_6 to each witness.

$$m_5 = E_{k_{SU_j,BS_i}}\left(IND_P \| k_{W_1,SU_j} \| k_{W_2,SU_j} \| \ldots \| k_{W_N,SU_j} \| t_5\right) \| MAC_{k_{SU_j,BS_i}}\left(m_5'\right), \tag{5}$$

$$m_6 = E_{k_{W_k,BS_i}}\left(IND_P \parallel n_i \parallel k_{W_k,SU_j} \parallel t_6\right) \parallel MAC_{k_{W_k,BS_i}}\left(m_6'\right), \tag{6}$$

where $m_5' = (IND_P \parallel k_{W_1,SU_j} \parallel k_{W_2,SU_j} \parallel \ldots \parallel k_{W_N,SU_j} \parallel t_5)$ and $m_6' = (IND_P \parallel n_i \parallel k_{W_k,SU_j} \parallel t_6)$. BS_i generates a table for this location proof and verification, as shown in Table 2.

(3) After receiving the message m_6, a selected witness generates a random number r_i and broadcasts the following message m_7 through its predefined short-range communication interface for a period of time T.

$$m_7 = n_i \parallel IND_P \parallel t_7 \parallel E_{k_{W_k,SU_j}}(n_i \parallel IND_P \parallel r_i \parallel t_7) \parallel MAC_{k_{W_k,SU_j}}\left(m_7'\right), \tag{7}$$

where $m_7' = (n_i \parallel IND_P \parallel r_i \parallel t_7)$. Another r_i is generated and broadcast in a similar way. This process is repeated until the witness receives a response from SU_j.

(4) SU_j first ensures that the IND_P is the same as the one already received from the BS when the SU receives the message m_7. Then, SU_j checks whether t_7 is fresh and verifies $MAC_{k_{W_k,SU_j}}(m_7')$. If these verifications are not sustained, SU_j drops this message and waits for the next message. Otherwise, SU_j decrypts the message with the key k_{W_k,SU_j} that matches n_i, and verifies whether or not the message has been tampered with. If the message cannot be decrypted by the corresponding key, SU_j discards this message. If all the verifications are successful, SU_j must immediately send message m_8 to W_k:

$$m_8 = IND_P \parallel t_8 \parallel E_{k_{W_k,SU_j}}\left(IND_P \parallel r_i \parallel t_8 \parallel H\left(PID_{SU_i} \parallel r_i\right)\right) \parallel MAC_{k_{W_k,SU_j}}\left(m_8'\right), \tag{8}$$

where $m_8' = (IND_P \parallel r_i \parallel t_8 \parallel H(PID_{SU_i} \parallel r_i))$.

(5) Upon receiving m_8, W_k first checks whether IND_P in m_8 is the same as the IND in m_6. Then, W_k verifies $MAC_{k_{W_k,SU_j}}(m_8')$ and checks whether r_i in m_8 is the last random number that had been broadcast by itself. If so, W_k generates the following location proof LP and sends it to BS_i.

$$LP = n_i \parallel IND_P \parallel t_9 \parallel E_{k_{W_k,BS_i}}\left(IND_P \parallel PID_{W_k} \parallel Loc_{W_k} \parallel r_i \parallel t_9 \parallel H\left(PID_{SU_j} \parallel r_i\right)\right) \parallel MAC_{k_{W_k,BS_i}}\left(LP'\right), \tag{9}$$

where $LP' = (IND_P \parallel PID_{W_k} \parallel Loc_{W_k} \parallel r_i \parallel t_9 \parallel H(PID_{SU_j} \parallel r_i))$. Otherwise, the following *null LP* is sent to BS_i:

$$nLP = n_i \parallel IND_P \parallel t_{10} \parallel E_{k_{W_k,BS_i}}\left(IND_P \parallel PID_{W_k} \parallel null \parallel t_{10}\right) \parallel MAC_{k_{W_k,BS_i}}\left(nLP'\right), \tag{10}$$

where $nLP' = (IND_P \parallel PID_{W_k} \parallel null \parallel t_{10})$.

4.5. Location Verification Phase. For each *non – null LP* that has been received before the time threshold, BS_i checks *non – null LPs* as follows.

(1) Checking whether IND_P is the unique IND for this location proof, and finding out PID_{SU_j} in the message m_3 according to IND_P.

(2) Checking whether the time stamp t_9 is fresh.

(3) According to n_i, trying to find the unique session key k_{W_k,BS_i}, and using k_{W_k,BS_i} to decrypt LP. Then, BS_i checks whether PID_{W_k} is the selected witness.

(4) Computing $MAC_{k_{W_k,BS_i}}(LP')$ and verifying it is the same as $MAC_{k_{W_k,BS_i}}(LP')$ received through LP.

(5) Verifying whether Loc_{SU_j} is in an acceptable range of Loc_{W_k}.

(6) Computing $H(PID_{SU_j} \parallel r_i)$ and verifying it is the same as $H(PID_{SU_j} \parallel r_i)$ received through LP.

(7) Checking whether the number of *non – null LPs* is greater than a predefined threshold N_H.

If all the above checks are successful, BS_i accepts SU_j, which means that SU_j is legitimate. Otherwise, SU_j's request is rejected.

4.6. Channel Allocation Phase. If the above location verification is successful, which means SU_j is where he/she claims to be, BS_i chooses a channel from the channel list, which is preallocated from the DB. One crucial step for an SU to protect location privacy and to get a better service quality, is to choose the most stable channel. The available channels in the channel list can be divided into four cases and are analyzed below.

(1) All PUs operating on these channels are on-line but close to SU_j. The BS should choose the channel with the higher power for the SU, so that the SU can obtain the best quality of service.

(2) Some PUs in this case might be off-line. BS_i chooses the channel with the highest power, but not the maximum power, and long usage time for SU_j. Although in this case, SU_j can obtain the channels where the PUs are off-line and get the best service quality, once the PU goes on-line, the DB will force SU_j to go off-line and switch channels. Therefore, SU_j may be caught in frequent channel switching.

(3) All PUs operating on these channels are on-line but far away from SU_j. This kind of channel is preferred to be used, so BS_i chooses this channel for SU_j. When SU_j accesses such channel, SU_j can work at the maximum power regardless of whether the PU is on-line or off-line, so that SU_j can use the channel with the best quality of service and avoid the privacy leaking that can be caused by channel switching.

(4) Some PUs might be off-line while other PUs are on-line but might be far away from SU_j. BS_i should choose the channel with the highest (but not the maximum) power and long usage time for SU_j, ensuring that SU_j can get the best quality channel.

As analyzed above, the power and the channel usage time are determined by the channel quality of service. BS_i should choose the channel with the best quality of service for SU_j

Table 2: The information of this location proof and verification.

IND	PID_{SU}	$k_{SU,BS}$	PID_W	$k_{W,BS}$	$k_{W,SU}$	t
IND_P	PID_{SU_j}	k_{SU_j,BS_i}	$PID_{W_1} \parallel \ldots \parallel PID_{W_N}$	$k_{W_1,BS_i} \parallel .. \parallel k_{W_N,BS_i}$	$k_{W_1,SU_j} \parallel \ldots \parallel k_{W_N,SU_j}$	t_5

according to the power and the channel usage time. After choosing the best channel, BS_i sends the message Quc to SU_j.

$$Quc = E_{k_{SU_j,BS_i}}(Ch_i \parallel t_{11}) \parallel MAC_{k_{SU_j,BS_i}}(Quc'), \quad (11)$$

where $Quc' = (Ch_i \parallel t_{11})$. When multiple SUs need the same channel at the same time, the BS preferentially allocates this channel to the SU with a high reputation value.

When SU_j plans to go off-line, SU_j sends the message Bye to BS_i.

$$Bye = E_{k_{SU_j,BS_i}}(0 \parallel t_{12} \parallel PID_{SU_j}) \parallel MAC_{k_{SU_j,BS_i}}(Bye'), \quad (12)$$

where $Bye' = (0 \parallel t_{12} \parallel PID_{SU_j})$. After receiving the message Bye, the BS reports to the TA, and the TA deletes PID_{SU_j} that SU_j has used.

4.7. Reputation Update Phase. As each SU sends his/her real identity to the TA for registration, the TA stores the mappings between the pseudo-IDs and the real identity of each SU. After receiving messages from each BS, the TA constructs a table to record the location verification results of each SU, as shown in Table 3. In Table 3, R_{SU_j} indicates whether BS_i accepts SU_j; 0/1 means BS_i rejects/accepts SU_j. In addition, WR_{SU_j} indicates whether SU_j provides an effective location proof as a selected witness; 0/1 means SU_j has provided an effective/invalid location proof as a selected witness.

From this table, the TA can trace each SU and check the honesty of each SU, for instance, whether an SU is honest in claiming his/her location or has provided effective location proof for the surrounding SUs as witness.

After tracing each SU, the TA updates the reputation value Rep_{SU_j} of SU_j in Table 4 as

$$Rep_{SU_j} = \frac{C_{R_{SU_j=1}+WR_{SU_j=1}}}{C_{SU_j}}, \quad (13)$$

where $R_{SU_j=1}$ means R_{SU_j}'s value that is equal to 1 and $WR_{SU_j=1}$ means WR_{SU_j}'s value that is equal to 1 in Table 3. And $C_{R_{SU_j=1}+WR_{SU_j=1}}$ is the sum of the numbers of $R_{SU_j=1}$ and $WR_{SU_j=1}$, and C_{SU_j} is the sum of the numbers of R_{SU_j} and WR_{SU_j}. There are three cases.

(1) $Rep_{SU_j} = (C_{R_{SU_j=1}+WR_{SU_j=1}} + 1)/(C_{SU_j} + 1)$: SU_j has honestly claimed his/her location or SU_j has provided the effective location proof for the surrounding SUs as witness.

(2) $Rep_{SU_j} = (C_{R_{SU_j=1}+WR_{SU_j=1}} - 1)/(C_{SU_j} + 1)$: SU_j has been rejected by the BS or SU_j has not provided the effective location proof for the surrounding SUs as a witness.

(3) $Rep_{SU_j} = Rep_{SU_j}$: During this update period, SU_j has not sent the channel request to the BS or SU_j has not been selected as a witness.

5. Security Analysis

In this section, we analyze the security of the proposed protocol with respect to the security requirements given in Section 3.2.

Privacy preservation: In our protocol, the SUs obtain pseudo-IDs in the system initialization phase. The SUs use these pseudo-IDs instead of real identities at the channel query phase. Except for the TA, anyone (including the DB, the BSs, and other SUs) cannot obtain the SUs' real identities. However, in our protocol, the SUs do not provide their location information for the DB for a channel request, so that the DB has no knowledge about the location information of SUs. Further, since there are only pseudo-IDs stored in each BS, the BS can only obtain the current location of SUs. The pseudo-IDs will change at the next channel request. Therefore, the BS cannot obtain the trajectory of SUs. Even if all BSs have been attacked, the attacker cannot map between the real identity and the location information of SUs. That is, our protocol can prevent anyone from discovering the real identity and location information of SUs.

Location verification: Each SU who wants to obtain the available channels must be verified by the BS. The BS selects some surrounding SUs to be the witnesses to verify the SU's location. The location proofs generated by the witness are unforgeable. We consider several scenarios. If a malicious SU wants to generate a location proof by himself, the BS will detect this: the BS receives location proofs from selected witnesses. The BS checks the received LPs and decrypts the LPs by the unique session key $k_{W_k,BS}$. Since any entity does not share his/her private key, this SU cannot generate a LP even if he knows the identity of each selected witness. Malicious SUs who try to send fake location information to the BS or to generate a location proof by himself will be rejected by BS.

Attack resistance: Since all messages have been encrypted and protected by secret keys, even though an attacker can capture all messages transmitted between each entity, it cannot acquire the content of messages. We use a message authentication code to prevent messages from being tampered with by attackers. Anyone who wants to change the information in these messages will first need to be verified, and fake messages will be removed. Meanwhile, we use time stamps to prevent the attack from being replayed, so such an attack is infeasible in our protocol.

Resistance to SU-Witness collusion: At the location verification phase, the BS checks all LPs and rejects any LP which is not generated by selected witnesses. Thus, a malicious

Table 3: The information of each SU in the TA.

ID_{SU}	PID_{SU}	R_{SU_j}	WR_{SU_j}	Record time
ID_{SU_1}	$PID_{SU_{1,1}}, \ldots, PID_{SU_{1,x}}$	$1 \parallel 1 \parallel \ldots$	$1 \parallel 1 \parallel \ldots$	t_1
ID_{SU_2}	$PID_{SU_{2,1}}, \ldots, PID_{SU_{2,x}}$	$1 \parallel 0 \parallel \ldots$	$1 \parallel 1 \parallel \ldots$	t_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

Table 4: The reputation value of each SU.

ID_{SU}	PID_{SU}	Reputation	Update time
ID_{SU_1}	$PID_{SU_{1,1}}, \ldots, PID_{SU_{1,x}}$	Rep_{SU_1}	t_1
ID_{SU_2}	$PID_{SU_{2,1}}, \ldots, PID_{SU_{2,x}}$	Rep_{SU_2}	t_1
$\ldots$	$\ldots$	$\ldots$	$\ldots$

SU cannot generate a *LP* for himself or any other users. This makes it very difficult for a malicious SU to set up successful SU-Witness collusion. Each selected witness has a high reputation value, which means they have a high probability of being an honest SU and will not collude with a malicious SU. Reputation values motivate them to generate efficient *LPs* since reputation values will affect the service quality of the channel they receive from the BS. However, we cannot altogether exclude the possibility of SU-Witness collusion. Any malicious SU needs to increase the size of his/her collusion group to improve his/her chances of success. In other words, the number of non-null *LPs* should be greater than a predefined threshold N_T. It means unless an SU colludes with at least N_T selected witnesses, it cannot succeed. However, in our protocol the BS selects witnesses: this selection mechanism ensures that the number of dishonest witnesses is less than N_T. Therefore, the possibility of SU-Witness collusion is minuscule in the proposed protocol.

Comparison with other protocols: We compare the security of the proposed protocol with protocols discussed in Section 2. We also give an overview of the related literature and summarize the comparison between these protocols and our protocol in Table 5. Table 5 shows that our protocol is more secure than other protocols.

6. Performance Evaluation

In this section, we evaluate the performance of our proposed protocol, by showing its computation and communication costs, and compare them with the protocols proposed by Grissa et al. [21] and Gao et al. [20]. All experiments have been conducted on a 64-bit computer with an Intel Core i7 CPU of 2.5 GHz and 16G memory. At the channel preallocation phase, a BS encrypted m_1 with AES with 256-bit key size (AES-256) and the TA also encrypts m_2 with AEC-256. At the channel query phase, the channel request message from an SU to a BS is also encrypted with ECC-256. In other phases, the messages transmitted between a BS and an SU, an SU and witnesses, and witnesses and a BS are encrypted with AES with 256-bit key size.

It is assumed that the network is divided into $m \times m$ cells; the output of hash function is 160 bits; ID_{BS_i},PID_{SU_j}, PID_{W_k}, *timestamp*, and channel list are 32 bits, respectively; a random number r_i is 128 bits; and Loc_{BS_i}, Loc_{SU_j}, and Loc_{W_k} are 40 bits, respectively. Further, let $T_{EC}, T_{DC}, T_{EA}, T_{DA}, T_H$, and T_{MAC} denote the running time for one ECC-256 encryption operation, one ECC-256 decryption operation, one AES-256 encryption operation, one AES-256 decryption operation, one hash function operation (SHA1 on 512-bit block), and one message authentication code operation (SHA1 on 512-bit block), respectively. The running time of $T_{EC}, T_{DC}, T_{EA}, T_{DA}, T_H$, and T_{MAC} is $3.124ms$, $2.926ms$, $0.006ms$, $0.006ms$, $0.001ms$, and $0.003ms$, respectively.

6.1. The Performance Analysis of Our Protocol. In this subsection, we analyze the computation and communication cost of channel preallocation phase (CPAP), channel query phase (CQP), location proof generation phase (LPGP), location verification phase (LVP), and channel allocation phase (CAP) in our protocol in Table 6.

6.1.1. Computation Cost. We analyze the computation cost of each phase.

(1) CPAP: At the channel preallocation phase, each BS encrypts m_1' and sends m_1 to the DB. Each BS executes one AES encryption operation and one message authentication code operation. To allocate the channel list to the BS, the DB needs to execute one AES encryption operation to encrypt m_2'. After receiving m_2, each BS also needs to execute one AES decryption operation to decrypt m_2 and obtains the channel list. In this phase, the computation cost of the BS is $T_{EA} + T_{DA} + 2T_{MAC}$ and the computation cost of the DB is $T_{EA} + T_{DA} + 2T_{MAC}$.

(2) CQP: At the channel query phase, the computation cost of the SU is one elliptic curve encryption operation and one message authentication code operation, which is $T_{EC} + T_{MAC}$. The computation cost of the BS is one elliptic curve decryption operation and one message authentication code operation, which is $T_{DC} + T_{MAC}$.

(3) LPGP: At the location proof generation phase, the computation cost of each BS is $N + 1$ AES encryption operations and $N+1$ message authentication code operations, which is $(N + 1) \cdot (T_{EA} + T_{MAC})$. The computation cost of

Table 5: Comparison of existing privacy-preserving and location verification protocols in CRNs.

Protocol	Technique	Privacy protection	Location verification	Collusion resistant
Gao et al. [20]	PIR	√	×	×
Troja et al. [12]	PIR	√	×	×
Grissa et al. [21]	Cuckoo Filter	√	×	×
Petrov et al. [13]	k-anonymity	√	×	×
Xin et al. [16]	PIR and APs-based	√	√	×
Li et al. [4]	Cryptography and APs-based	√	√	×
Our protocol	Cryptography and distributed	√	√	√

Table 6: Computation and communication costs of our protocol.

Phase	Entity	Computation cost	Communication cost (bits)
CPAP	DB	$T_{EA} + T_{DA} + 2T_{MAC}$	288
	BS	$T_{EA} + T_{DA} + 2T_{MAC}$	328
CQP	BS	$T_{DC} + T_{MAC}$	0
	SU	$T_{EC} + T_{MAC}$	1440
LPGP	TA	0	$(K+1)\cdot 34$
	BS	$(N+1)\cdot(T_{EA}+T_{MAC})$	$(K+1)\cdot 32 + 738N + 224$
	SU	$N\cdot T_{EA} + (N+1)\cdot T_{DA} + (2N+1)\cdot T_{MAC} + N\cdot T_H$	$608N$
	Witnesses	$(n+1)\cdot N\cdot T_{EA} + 2N\cdot T_{DA} + (n+3)\cdot N\cdot T_{MAC}$	$452\cdot n\cdot N + 682Nbits$
LVP	BS	$N\cdot(T_{DA}+T_{MAC}+T_H)$	0
CAP	BS	$T_{EA} + T_{MAC}\cdot$	224
	SU	$T_{DA} + T_{MAC}$	0

each SU consists of N AES encryption operations, $N+1$ AES decryption operations, $2N+1$ message authentication code operations, and N hash function operations, which is $N\cdot T_{EA} + (N+1)\cdot T_{DA} + (2N+1)\cdot T_{MAC} + N\cdot T_H$ (N denotes the number of selected witnesses). Similarly, the computation cost of witnesses consists of $(n+1)\cdot N$ AES encryption operations (n denotes the number of broadcasted rounds), $2N$ AES decryption operations, and $(n+3)\cdot N$ message authentication code operations, which is $(n+1)\cdot N\cdot T_{EA} + 2N\cdot T_{DA} + (n+3)\cdot N\cdot T_{MAC}$.

(4) LVP and CAP: In the location verification phase and the channel allocation phase, the computation cost of each BS is N AES decryption operations, $N+1$ message authentication code operations, N hash function operations, and one AES encryption operation, which is $N\cdot(T_{DA}+T_H)+(N+1)\cdot T_{MAC}+T_{EA}$. The SU needs one AES decryption operation and one message authentication code operation, which is $T_{DA}+T_{MAC}$.

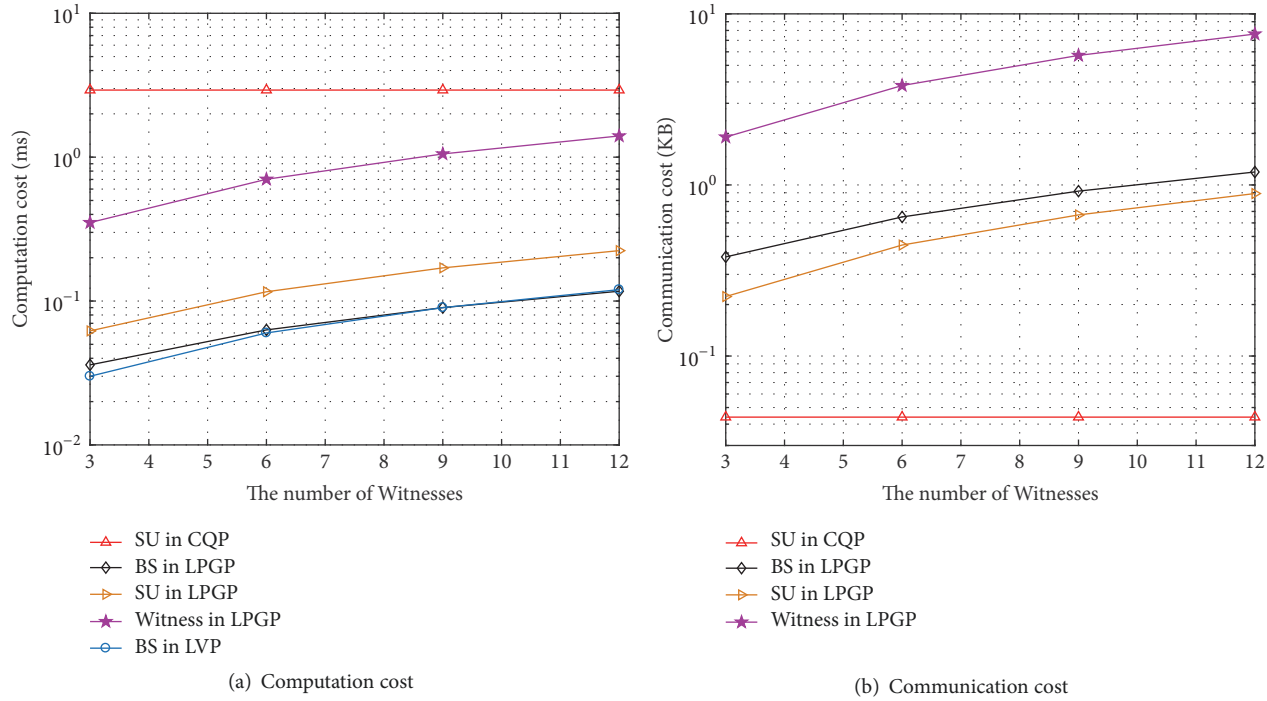
6.1.2. Communication Cost. We analyze the communication cost of each phase.

(1) CPAP: At the channel preallocation phase, each BS sends the message $m_1 = ID_{BS_i} \parallel t_1 \parallel E_{k_{DB,BS_i}}(ID_{BS_i} \parallel Loc_{BS_i} \parallel t_1) \parallel MAC_{k_{DB,BS_i}}(m_1')$ to the DB, and the DB chooses a channel list and sends the message $m_2 = ID_{BS_i} \parallel t_2 \parallel E_{k_{DB,BS_i}}(Ch_{BS_i} \parallel t_2) \parallel MAC_{k_{DB,BS_i}}(m_2')$ to the BS. Hence, the communication cost of the BS is $(32+32+32+40+32+160) = 328\ bits$, and the communication cost of the DB is $(32+32+32+32+160) = 288\ bits$.

(2) CQP: At the channel query phase, each SU sends the channel request message $m_3 = E_{PK_{BS_i}}(PID_{SU_j} \parallel Loc_{SU_j} \parallel k_{SU_j,BS_i} \parallel t_3)$ to the BS. Therefore, the communication cost is $(32+40+256+32)\times 4 = 1440\ bits$.

(3) LPGP: At the location proof generation phase, the TA responds to the reputation value of the K SUs surrounding the SU, so that the communication cost is $(K+1)\cdot 34$. The BS sends the reputation value request message to the TA, $m_5 = E_{k_{SU_j,BS_i}}(IND_P \parallel k_{W_1,SU_j} \parallel k_{W_2,SU_j} \parallel \cdots \parallel k_{W_N,SU_j} \parallel t_5) \parallel MAC_{k_{SU_j,BS_i}}(m_5')$ to the SU, and N $m_6 = E_{k_{W_k,BS_i}}(IND_P \parallel n_i \parallel k_{W_k,SU_j} \parallel t_6) \parallel MAC_{k_{W_k,BS_i}}(m_6')$ to each witness, respectively. The communication cost of the BS is $(K+1)\times 32 + 32 + 256N + 32 + 160 + (32+32+2+256+160)\times N = (K+1)\cdot 32 + 738N + 224\ bits$. The SU responds to the message $m_8 = IND_P \parallel t_8 \parallel E_{k_{W_k,SU_j}}(IND_P \parallel r_i \parallel t_8 \parallel H(PID_{SU_i} \parallel r_i)) \parallel MAC_{k_{W_k,SU_j}}(m_8')$ to each witness, so that the communication cost is $N\times(32+32+32+160+32+160+160) = 608Nbits$. Each witness broadcasts n rounds of the message $m_7 = n_i \parallel IND_P \parallel t_7 \parallel E_{k_{W_k,SU_j}}(n_i \parallel IND_P \parallel r_i \parallel t_7) \parallel MAC_{k_{W_k,SU_j}}(m_7')$ and sends $LP = n_i \parallel IND_P \parallel t_9 \parallel E_{k_{W_k,BS_i}}(IND_P \parallel PID_{W_k} \parallel Loc_{W_k} \parallel r_i \parallel t_9 \parallel H(PID_{SU_j} \parallel r_i)) \parallel MAC_{k_{W_k,BS_i}}(LP')$ to the BS, respectively. Hence, the communication cost of witnesses is $(n\times N\times(2+32+32+32+32+2+160+160)) + N\times(2+32+32+32+32+40+160+32+160+160) = 452\cdot n\cdot N + 682N\ bits$.

(4) LVP and CAP: At the location verification phase and the channel allocation phase, the BS chooses a channel



(a) Computation cost

(b) Communication cost

Figure 2: Computation cost and communication cost of our protocol.

and sends the message $Quc = E_{k_{SU_j,BS_i}}(Ch_i \parallel t_{11}) \parallel MAC_{k_{SU_j,BS_i}}(Quc')$ to the SU, so that the communication cost is $(32 + 32 + 160) = 224$ *bits*.

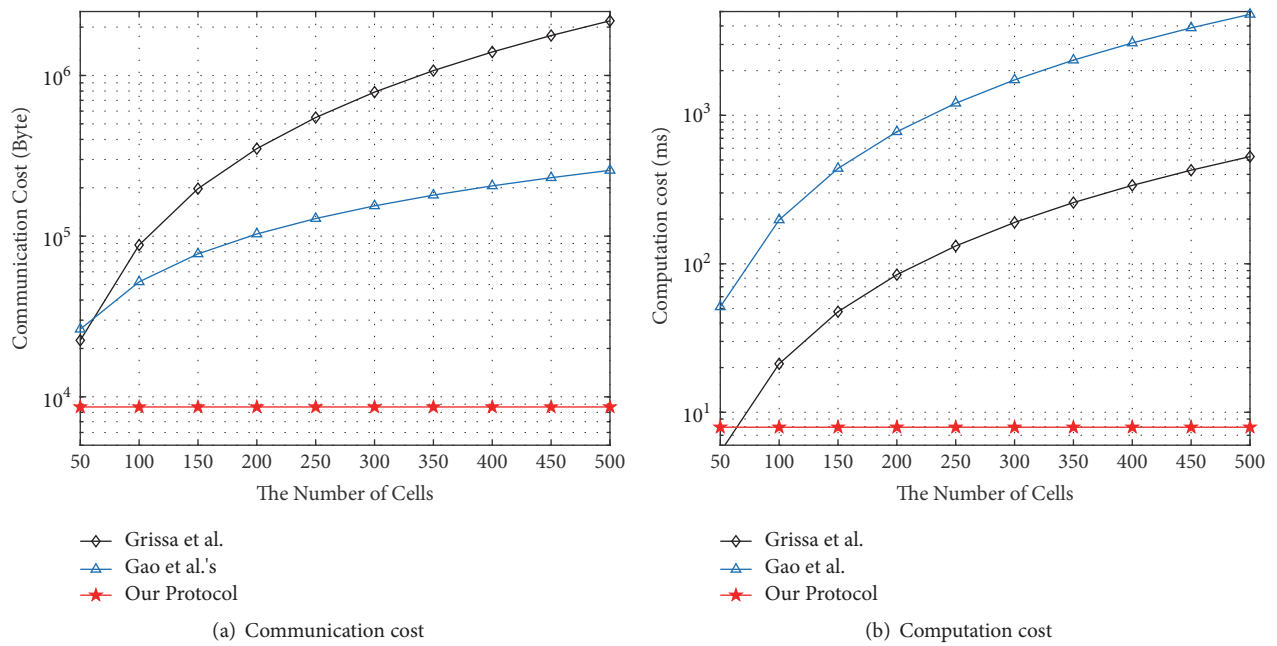
Moreover, the computation and communication cost of the different number of witnesses is shown in Figure 2 using the expressions established in Table 6, wherein K is set to be 20 and n is set to be 10. In order to show our communication and computational costs clearly, we use a logarithmic way of drawing. As shown in Figure 2(a), when the number of witnesses is 3, the running time of the BS and that of the DB at the channel preallocation phase are less than 1 *ms*; the running time of an SU obtaining the location proof at the location proof generation phase is less than 0.4 *ms*, and the running time of a BS to verify the location and to allocate a channel to an SU at the location verification and channel allocation phases is less than 0.05 *ms*. Even though the number of witnesses is 12, the running time of an SU obtaining the location proof at the location proof generation phase and obtaining a channel in location verification phase and channel allocation phase is less than 1.4 *ms*. As shown in Figure 2(b), when the number of witnesses is 3, the communication cost of the BS and that of the DB in channel preallocation phase are about 77 *bytes*, and the communication cost of an SU obtaining the location proof at the location proof generation phase is about 228 *bytes*; the communication cost of a BS to verify the location and to allocate a channel to an SU at the location verification and channel allocation phases is about 8 *bytes*. Although the number of witnesses is 12, the communication cost of an SU obtaining the location proof in that and the subsequent phases and obtaining a channel is about 912 *bytes*.

6.2. Comparison with Other Protocols. In this subsection, we compare the performance of the proposed protocol with other protocols proposed by Gao et al. [20] and Grissa et al. [21], and we provide analytical expressions of the communication of these protocols in Table 7 and computation cost of these protocols in Table 8. We plot the comparison of computation and communication costs of these protocols in Figure 3 using the expressions established in Tables 7 and 8. Figure 3(a) shows that our protocol is superior to the protocols proposed by both Gao et al. [20] and Grissa et al. [21] in communication costs, because we used symmetric encryption operations in our protocol, while Figure 3(b) shows the comparison of the computation cost of these protocols. As shown in Figure 3(b), when the number of cells increases, our computation cost remains 1.4 ms, which proves our protocol is lightweight in computation cost and unaffected by of the number of cells.

In summary, Figure 3 shows the performance of our protocol is superior to that of the protocols proposed by Gao et al. [20] and Grissa et al. [21] and is unaffected by the number of cells. Meanwhile, the performance analysis also shows our protocol is lightweight in both computation and communication costs, which achieves a much better performance in comparison with other protocols.

7. Conclusion

In this paper, we propose a lightweight privacy-preserving location verification protocol (LPPLV) for large-scale database-driven CRNs. In LPPLV, the SU does not need to provide his/her real identity and location information



(a) Communication cost

(b) Computation cost

Figure 3: Comparison of computation cost and communication cost.

Table 7: Communication cost of different protocols.

	Communication cost
Gao et al. [20]	$(2m+3)\cdot[\log p]$
Grissa et al. [21]	$\sigma_{query}+\varphi\cdot s\cdot m^2\cdot\dfrac{(\log_2(1/\epsilon)+\log_2(2\delta))}{\delta}+s\cdot\sigma_{HMAC}$
Our protocol	$2306+66K+1548N+292\cdot n\cdot N$

Table 8: Computation costs of different protocols.

	Computation cost		
	SU	BS	DB
Gao et al. [20]	$4m\cdot T_M$	-	$O(m^2)\cdot T_M$
Grissa et al. [21]	$s\cdot T_H$	$3s\cdot T_H$	$3\varphi\cdot s\cdot m^2\cdot T_H$
Our protocol	$T_{EC}+(N+2)\cdot T_{DA}+N\cdot T_{EA}+N\cdot T_H+(2N+3)\cdot T_{MAC}$	$(N+3)\cdot T_{EA}+N\cdot T_H+(N+1)\cdot T_{DA}+T_{DC}+(2N+5)\cdot T_{MAC}$	$T_{EA}+T_{DA}+2T_{MAC}$

for the DB, which prevents the DB from obtaining the real identity and location information of SUs. LPPLV has a distributed architecture in which SUs generate location proofs for each other. The new witness selection mechanism allows malicious SUs to be discovered by the BS when malicious SUs generate fake location information. This new witness selection mechanism also provides a solution for possible SU-Witness collusion, which has not been well solved in existing works. Further, by channel preallocation, LPPLV improves the efficiency of obtaining available channels for SUs. The results of security analysis prove that LPPLV resists various types of attacks in database-driven CRNs. Experimental results indicate the efficiency of LPPLV.

In future work, we will study how to improve the security and privacy of PUs. Because SUs can learn the state of PUs from the channel, this might be easily misused and threaten the privacy of PUs. Therefore, we will take the PUs' privacy into account in our future work.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China [Grant No. 61771140, No.61841701], Major Program of Fuzhou Municipal Bureau of Science and Technology [Grant No. (2017)325], Major Science and Technology Project in Fujian Province [Grant No. 2017H6005], and Natural Science Foundation of Fujian Province (Grant No.2018J01560).

References

[1] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public cloud systems," *IEEE Systems Journal*, pp. 1–9.

[2] Y. Li, L. Zhou, H. Zhu, and L. Sun, "Privacy-preserving location proof for securing large-scale database-driven cognitive radio networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 563–571, 2016.

[3] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: a protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.

[4] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the internet of things in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 84–91, 2017.

[5] Y. Sun and K. Chowdhury, "Enabling emergency communication through a cognitive radio vehicular network," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 68–75, 2014.

[6] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: a survey," *Physical Communication*, vol. 4, no. 1, pp. 40–62, 2011.

[7] M. Grissa, B. Hamdaoui, and A. A. Yavuza, "Location privacy in cognitive radio networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1726–1760, 2017.

[8] "Federal Communications Commission," http://transition.fcc.gov/DailyReleases/DailyBusiness/2012/db0405/FCC-12-36A1.pdf, [Last accessed May 2012].

[9] Y. Zeng, L. Xu, X. Yang, and X. Yi, "An efficient privacy-preserving protocol for database-driven cognitive radio networks," *Ad Hoc Networks*, pp. 1–11, 2018.

[10] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Cuckoo filter-based location-privacy preservation in database-driven cognitive radio networks," in *Proceedings of the IEEE Computer Networks and Information Security on Proceedings, 2015 World Symposium on*, pp. 1–7, Hammamet, Tunisia, 2015.

[11] S. Pontarelli, P. Reviriego, and J. Maestro, "Parallel d-pipeline: a cuckoo hashing implementation for increased throughput," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 1, pp. 326–331, 2016.

[12] E. Troja and S. Bakiras, "Leveraging P2P interactions for efficient location privacy in database-driven dynamic spectrum access," *International Journal of Network Security*, vol. 17, no. 5, pp. 569–579, 2015.

[13] D. Petrov and T. Znati, "Location privacy preserving protocols in database-enabled cognitive radio networks," in *Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC) on Proceedings,13th International*, pp. 147–152, IEEE, Spain, 2017.

[14] L. Zhang, C. Fang, Y. Li, H. Zhu, and M. Dong, "Optimal strategies for defending location inference attack in database-driven CRNs," in *Proceedings of the IEEE International Conference on Communications, ICC on Proceedings*, pp. 7640–7645, IEEE, Londen, UK, 2015.

[15] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. J. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *Proceedings of the Dynamic Spectrum Access Networks (DYSPAN) on Proceedings, 2014 IEEE International Symposium on*, pp. 236–247, IEEE, McLean, VA, USA, 2014.

[16] J. Xin, M. Li, C. Luo, and P. Li, "Privacy-preserving spectrum query with location proofs in database-driven CRNs," in *Proceedings of the IEEE Global Communications Conference on Proceedings*, pp. 1–6, IEEE, Washington, DC, USA, 2016.

[17] M. Reza Nosouhi, S. Yu, M. Grobler, Y. Xiang, and Z. Zhu, "SPARSE: privacy-aware and collusion resistant location proof generation and verification," in *Proceedings of the GLOBECOM 2018 - 2018 IEEE Global Communications Conference*, pp. 9–13, IEEE, Abu Dhabi, UAE, 2018.

[18] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M. Killijian, "Xpir: private information retrieval for everyone," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 2, pp. 155–174, 2016.

[19] Y. Zhang, W. Tong, and S. Zhong, "On designing satisfaction-ratio-aware truthful incentive mechanisms for k-anonymity location privacy," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2528–2541, 2016.

[20] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: attacks and countermeasures," in *Proceedings of the IEEE International Conference on Computer Communications on Proceedings*, pp. 2751–2759, IEEE, Turin, Italy, 2013.

[21] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Location privacy preservation in database-driven wireless cognitive networks through encrypted probabilistic data structures," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 255–266, 2017.

[22] X. Wang, A. Pande, J. Zhu, and P. Mohapatra, "STAMP: enabling privacy-preserving location proofs for mobile users," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3276–3289, 2016.

[23] Z. Zhu and G. Cao, "APPLAUS: a privacy-preserving location proof updating system for location-based services," in *Proceedings of the IEEE International Conference on Computer Communications on Proceedings*, pp. 1889–1897, Shanghai, China, 2011.

[24] C. Javali, G. Revadigar, K. B. Rasmussen, W. Hu, and S. Jha, "I am alice, i was in wonderland: secure location proof generation and verification protocol," in *Proceedings of the IEEE 41st conference on local computer networks (LCN) on Proceedings*, pp. 477–485, IEEE, Dubai, UAE, 2016.

[25] Y. Zhang, C. C. Tan, F. Xu, H. Han, and Q. Li, "VProof: Lightweight privacy-preserving vehicle location proofs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 378–385, 2015.

[26] Y. Zeng, X. Li, X. Yang, Q. Xu, and D. Wang, "A practical privacy preserving protocol in database-driven cognitive radio networks," in *Proceedings of the 23rd Australasian Conference on Information Security and Privacy(ACISP) on Proceedings*, pp. 634–648, Wollongong, Australia.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 8431074, 12 pages
https://doi.org/10.1155/2019/8431074

Research Article

Detection of Dummy Trajectories Using Convolutional Neural Networks

Jiaji Pan,[1] Yining Liu,[1] and Weiming Zhang[2]

[1]*School of Computer and Information Security, Guilin University of Electronic Technology, China*
[2]*CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei 230026, China*

Correspondence should be addressed to Yining Liu; lyn7311@sina.com

Received 25 January 2019; Revised 13 March 2019; Accepted 17 April 2019; Published 2 May 2019

Guest Editor: Fagen Li

Nowadays, privacy in trajectory is an important issue in the coming big data era. In order to provide better protection for trajectory privacy, a number of solutions have been proposed in the literature, and the dummy trajectory method has attracted great interests in both academia and industry recently due to the following advantages: (1) neither a third-party server nor other parties' cooperation is necessary; (2) location-based services are not influenced; and (3) its algorithm is relatively simple and efficient. However, most of trajectory privacy generations usually consider the geometric shape of the trajectory; meanwhile the real human mobility feature is usually neglected. In fact, the real trajectory is not the product of random probability. In this paper, convolutional neural network (CNN) is used as the learning machine to train with lots of the real trajectory and the generated dummy trajectory sets. Then, the trained classifier is used to distinguish the dummy from the real trajectory. Experiments demonstrate that the method using CNN is very efficient, and more than 90% of dummy trajectories can be detected. Moreover, the real trajectory erroneous judgment rate is below 10% for most of real trajectories.

1. Introduction

Nowadays, location-based services (LBS) [1–7] are widely used in smart mobile terminals, which makes the peoples' daily life more convenient. In the process of interaction between mobile terminal and LBS, there are many schemes to protect users' personal privacy data [8–10]. In addition, the large amount of trajectory data generated in the interaction process consists of the abundant space-time information such as users' personal interests, economic status, and living habits. In fact, this information is sensitive, and it should not be directly released to the public. On the other hand, the trajectory data is useful for the municipal transportation service, decision-making of government, and other business applications. Therefore, how to protect the data privacy [11, 12], especially to balance single trajectory privacy and the trajectory data publishing, is an interesting research topic, and it has attracted attentions all over the world.

The existing trajectory privacy protection methods include dummy trajectory method [13–16], trajectory suppression method [17], generalization method [6, 18], and differential method [19, 20].

For dummy trajectory method, several dummy trajectories are generated for each real trajectory; then the real trajectory and $k-1$ dummy trajectories are published together to reduce the probability of true trajectory exposure.

For trajectory suppression method, the sensitive information in the real trajectory is not released in order to protect the user's personal information contained in the trajectory, which needs to set or process the sensitive information in advance. Therefore, how to find the sensitive information becomes the key issue of the suppression method. Intuitively speaking, when there is a clear need to suppress the information, suppression method is simple and effective and can achieve the purpose of user privacy protection by simple data processing [17]. However, this method relies on the determination of sensitive information, that is, how to suppress the sensitive needs to know what resource is owned by the opponent. Obviously, it is not easy. In addition, the simple and crude direct deletion of sensitive information will reduce the usability of trajectory data.

For generalization method, its basic idea is to generalize the QI (Quasi identifier) attributes that can uniquely identify

the user, which guarantees that the real trajectory cannot be distinguished from other trajectories. The k-anonymity model is commonly used in trajectory privacy, which converts the D (trajectory data) in the database into D∗, so that any trajectory T in D∗ belongs to a trajectory k-anonymity set, and the information distortion between D∗ and D is minimized. In LBS, how to choose the scope of anonymous boxes is not easy, since the real trajectory is not known in advance.

For the differential method, it is used in trajectory privacy protection in recent years. It does not need to consider the background knowledge of the opponent and it is based on strict mathematical knowledge. It provides a quantifiable, assessable, and provable method for privacy protection. By adding random noise perturbation sensitive data, it can distort some data while maintaining its statistical properties.

In recent years, the dummy trajectory method is widely researched due to the following reasons:

(1) No third-party server is required, which makes it more robust.

(2) The algorithm of dummy trajectories generation is relatively simple and efficient.

(3) The service based on the precise location is not influenced since the real trajectory is kept.

There are many algorithms for generating dummy trajectory, such as rotation method, intraregion random point method, translation method, and the combination of these methods. The background factors are also taken into account in the generation of dummy trajectories. The basic idea of dummy trajectory was first proposed by Kido et al. [18, 21], in which there are two dummy trajectory design rules and the generation method. Lei et al. [14] proposed a method to increase the number of dummy trajectories by adding intersections to the trajectories obtained after rotation, therefore improving the privacy protection level of the real trajectory. In Wu et al.'s scheme [15], not only the distance between the real trajectory and the dummy trajectory is involved in the dummy trajectory generation, but also the distance between the dummy trajectories is also considered. By disturbing the generated dummy trajectories, the final set of trajectories can satisfy the privacy requirements. Kato R et al. [16] assumed that the user's movements are known in advance and proposed a dummy-based anonymization method based on the predicted movement, where dummies move naturally while stopping at several locations. Niu et al. proposed a dummy location generation algorithm based on background information, especially the probability of sending requests in each location being considered, and ensure that the generated $(k - 1)$ dummy locations can more easily confuse the opponent by formalizing the background information [22]. Hara et al. have further studied the problem of trajectory privacy protection in mobile vehicle network and designed a dummy trajectory generation algorithm based on vehicle trajectory [23]. Since the algorithm takes into account the trajectory characteristics of vehicle movement, the probability of dummy trajectory being guessed is reduced. Lu et al. [24] pointed out that although the generated dummy has a high density distribution, it can reduce the protection degree of users' location privacy. They divide the circle/grid region into subregions of equal size and distribute all positions over different radii/vertices. In literatures [25, 26], the authors focus on the real environment requirements considering physical constraints and propose a new virtual generation algorithm DumGrid and Dum-P [25]. Dum-P generates dummies around the user in grid mode; it ensures that the more realistic movement model generates dummies with the user's movement. To solve the problem of insufficient location privacy requirements, an improved version Dum-P-Cycle is proposed in [26]. In addition, perhaps chaotic system is a feasible tool to protect the trajectory information [27].

However, the human mobility model is not involved in most of the current dummy trajectory generation algorithms. In reality, the trajectory is not a set of random points but a set of points satisfying some known or unknown features, which is constrained by various conditions, especially people's behavior factors. For example, the deviation angle between the trajectory segment and the segment is usually very small (people tend to go straight); when the deviation angle suddenly increases (such as car turning), it means that the length of the following segments begins to decrease (such as the speed of driving turns decreased) and so on. In short, the trajectory points are arranged according to some certain rules, and these rules are often restrictive. Specifically, the real trajectory is usually purposeful. For the sake of efficiency, the trajectory segment consisting of the set of trajectory points generally has a small deflection angle and few frequent oscillations. People usually move with a uniform speed, so the length of trajectory segment should be gentle. However, most of the dummy trajectories generation relies on the use of a random method, the dummy trajectory deviation is frequent, and the length of the trajectory segment is oscillatory; therefore the trajectory points often fall in the nonreachable place. In conclusion, the current dummy trajectories generation algorithms do not take into account the behavioral characteristics contained in real trajectories; therefore there is a considerable probability of identifying dummy trajectories by analyzing the distribution characteristics of real trajectories.

In addition, there are complex laws between the points of real trajectories and dummy trajectories. The difference between real trajectory and dummy trajectory is difficult to be represented by simple function. Artificial neural network is a feasible tool, since it can be used as classifier in large-scale data training. Moreover, this process does not require much people interaction, and the parameters are automatically generated through a lot of iterative learning. As long as hyperparameters and network models are set reasonably, good classification results can be achieved.

In this paper, CNN model is used to generate the classification function, and the overall framework of our dummy trajectory detection method is shown in Figure 1.

We define a series of trajectory points to form a trajectory section, and two adjacent points form a trajectory segment. When a trajectory is detected, it is divided into sections of equal trajectory points at first. If the last section is short of the point number of trajectory section, the rounding method is adopted. In other words, less than half of the point number

$seg_i =< p_i, p_{i+1} > (i = 1, 2...k-1)$

$sec_i = \{seg_j, seg_{j+1}, ...seg_{j+m-1}\}\ \ (j = (i-1)\,m + 1)\ \left(i = 1, 2, ... \left\lfloor \frac{k}{m} \right\rfloor\right)$

If $(k\%m)/m >= 0.5$**:**

$sec_{\lfloor k/m \rfloor + 1} = \{seg_{k-m+1}, seg_{k-m+2}, ...seg_k\}$

Else: Round the rest segments

Algorithm 1

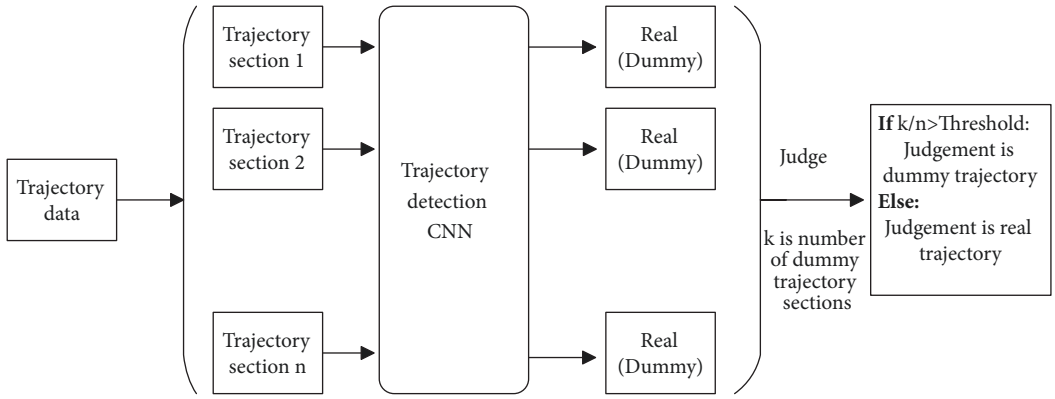


Figure 1: Dummy trajectory detection model framework based on CNN.

of trajectory section is rounded. Otherwise, the point is taken from the back to the front to reach the point number of trajectory section. For simplicity, in this paper, we suppose that a trajectory consists of m segments, and each m segment forms a section. The above concepts are defined as shown in Algorithm 1.

As shown (in which figure or algorithm), we obtain n trajectory sections; then we put n trajectory sections into the trained CNN, respectively, and obtain the judgment result of n trajectory sections. Suppose that k sections are judged as dummy; the trajectory is judged as the dummy when k/n is more than the predefined threshold. The lower the threshold is, the stricter the dummy trajectory detection is and the higher the dummy trajectory detection rate is but the higher the real trajectory erroneous judgment rate is.

The main contributions of our work are summarized as follows:

(1) Unlike previous works that set privacy standards and trajectory parameters to generate dummy trajectories, we try to find the differences between real trajectories and dummy trajectories from the attacker's point of view, which is useful for improving the dummy trajectory generation.

(2) The deep learning is used to train the behavior feature classifier of the human's movement; then the classifier is used to distinguish the dummy trajectories that are generated according to the current main algorithms.

2. Preliminaries

2.1. Trajectory Representation Method

Absolute Trajectory. The absolute trajectory consists of a series of trajectory points with latitude and longitude as the spatial metric, with time series as the trajectory points arrangement order, which is defined as $traj_{abs} = \{< loc_1(\mathrm{lat}_1,\mathrm{lng}_1),\mathrm{t}_1 >, < loc_2(\mathrm{lat}_2,\mathrm{lng}_2),\mathrm{t}_2 > ... < loc_n(\mathrm{lat}_n,\mathrm{lng}_n),\mathrm{t}_n >\}$. The trajectory data set released by major institutions is also an absolute trajectory.

Relative Trajectory. Although the absolute trajectory can accurately express the position of the trajectory on the earth's surface, it is not convenient to manipulate the trajectory such as stretching and rotating and to calculate some trajectory characteristics. Longitude and latitude can be regarded as absolute coordinates. The relative coordinate system describing the relative trajectory is a Cartesian coordinate system in which the point specified on the plane of the map where the trajectory is located is the coordinate zero and the direction specified is the x-axis and y-axis. In the process of trajectory data processing, we need to transform absolute trajectory into relative trajectory. The relative trajectory is defined as follows:

$$traj_{rel} = \{< loc_1(x_1,\mathrm{y}_1),\mathrm{t}_1 >, < loc_2(x_2,\mathrm{y}_2),\mathrm{t}_2 > ... < loc_n(x_n,\mathrm{y}_n),\mathrm{t}_n >\}$$

Feature Trajectory. Unlike the above two trajectory definitions, which use trajectory points to define trajectories, the feature trajectory is defined using trajectory shape features including the relative offset angles (roa) and relative lengths (rl) as follows:

$$traj_{fea} = \{< roa_1 = 0, rl_1, \mathrm{t}_1 >, < roa_2, rl_2, \mathrm{t}_2 > ... < roa_n, rl_n, \mathrm{t}_n >\}$$

2.2. Convolutional Neural Network. CNN is a kind of multilayer neural network, which is good at dealing with machine

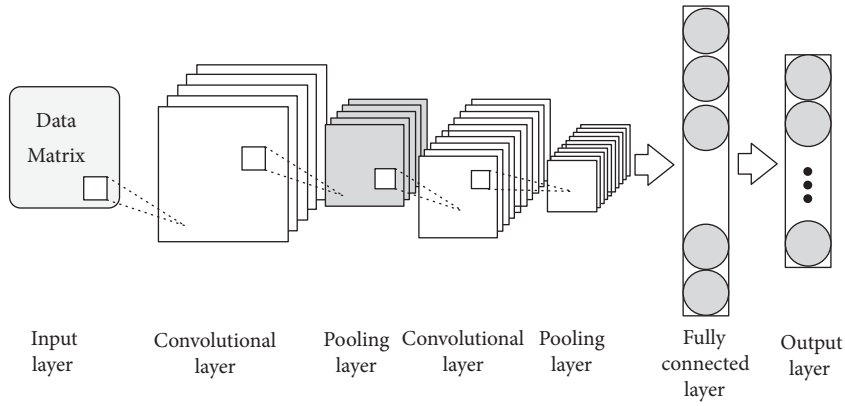


Figure 2: Classic CNN structure diagram.

learning problems related to images, especially large images [28]. CNN reduces the dimension of image recognition problem by a series of methods and makes it possible to be trained eventually. CNN consists of input layer, convolutional layer, activation function, pooling layer, and fully connected layer [29]. Inspired by the concept of local receptive field (participating in a convolutional kernel operation is an area of the input image (or feature map), the size of which is the receptive field), the convolutional layer connects the input small area and calculates the dot multiplication between the convolutional core and the corresponding input small area as the output. While the pooling layer is mainly designed to reduce the data dimension, it performs a downsampling operation on the spatial dimension (width, height). The classic CNN structure diagram is shown in Figure 2.

The convolutional layers are used for feature extraction, and we traverse the input matrix with a matrix called filter. The number of output matrices after each convolutional operation is the number of filters. The pooling layer compresses the input feature map. On one hand, it reduces the feature map and simplifies the network computing complexity; on the other hand, it compresses the feature and extracts the main features (we usually use maximum pooling). Fully connected layer connects all features and sends the output value to the classifier (such as Softmax classifier).

In summary, CNN extracts the features through convolutional layer and reduces the parameters and computational times through pooling layer. In fact, it completes classification tasks by traditional neural network. Compared with other classifiers, its filter used to extract data features has the characteristic of weight sharing. The adjacent trajectory feature metrics we need to extract have similar characteristics and we can share weights to extract. Therefore, CNN is compatible with our classification tasks.

3. Training and Detection

3.1. Feature Trajectory Definition

3.1.1. Relative Offset Angle. As shown in Figure 3, $\overrightarrow{a}$, $\overrightarrow{b}$, and $\overrightarrow{c}$ are three trajectory vectors and $\overrightarrow{d}$ is the extension line of vector $\overrightarrow{a}$. The vector angle θ_1 of vector $\overrightarrow{b}$ and vector $\overrightarrow{a}$ is defined as relative offset angle; also, the relative offset angle of $\overrightarrow{c}$ relative to $\overrightarrow{a}$ is θ_2. If $\overrightarrow{b}$ is over the vector $\overrightarrow{d}$, the offset angle is positive; otherwise it is negative.

3.1.2. Relative Length. We assume that the total length of the trajectory section is s, the i^{th} trajectory segment in the section is s_i, and the relative length of the i^{th} trajectory segment is defined as s_i/s.

3.2. Feature Trajectory Generation Algorithm. We use Algorithm 2 to generate the feature metrics of trajectory sections. No matter how the trajectory rotates or how the trajectory sections are scaled equally, the trajectory feature metric will not change. As long as the trajectory shape is the same, we regard it as the same trajectory. Similarly, for dummy trajectories, we also use this method to produce feature metrics of trajectory section. After generating the feature trajectories, we put the real and dummy trajectory set into the detector for training to initialize the detector.

3.3. Trajectory Data Preprocessing. In trajectory data preprocessing, the absolute trajectory is transformed to relative trajectory and then is transformed into feature trajectory. We take 5s as the time interval to extract a continuous series of points (longitude and latitude representation) from the testing trajectories. Then we use Algorithm 2 to transform relative trajectory (RT) into feature trajectory (FT). As shown in Table 1, we use RT and FT to represent a trajectory segment jointly.

3.4. Detector Design and Training Process

3.4.1. Detector Model Structure. There is a strong correlation between the trajectory segment and the trajectory segment; we use the correlation between the trajectory segments with CNN. Most of the other depth neural networks consider the data characteristics from the input data as a whole and cannot extract the trajectory characteristics very well. Compared with other artificial neural networks, the filter of CNN has unique weight sharing characteristics and we also need the sharing weights to extract feature of adjacent trajectory segments; therefore, CNN is used as classifier in

Input: k points intercepted on the trajectory (x_i, y_i), for i = 1 to k.
Output: k-2 relative offset angles (roa) and k-1 relative length (rl)

(1) $dis_{sum} = 0$
(2) **for** i= 1 to k-1 **do**
(3) $\overrightarrow{vec_i} = (x_{i+1} - x_i, y_{i+1} - y_i)$
(4) $dis_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
(5) $dis_{sum} += dis_i$
(6) **end for**
(7) **for** j = 1 to k-2 **do**
(8) $roa_j = \arccos\left(\dfrac{\overrightarrow{vec_{j+1}}\overrightarrow{vec_j}}{\left|\overrightarrow{vec_{j+1}}\right|\left|\overrightarrow{vec_{j+1}}\right|}\right)$
(9) $k = \dfrac{y_{j+1} - y_j}{x_{j+1} - x_j}$
(10) $b = y_j - kx_j$
(11) **if** $y_{j+2} > kx_{j+2} + b$
continue
(12) **else**
$roa = -roa$
(13) **end for**
(14) **for** t = 0 to k-1**do**
(15) $rl_t = \dfrac{dis_t}{dis_{sum}}$
(16) **end for**
(17) **Return** $traj_{fea} = \{< 0, rl_1 >, < roa_1, rl_2 > \ldots < roa_{k-2}, rl_{k-1} >\}$

Algorithm 2: Trajectory feature metrics generation.

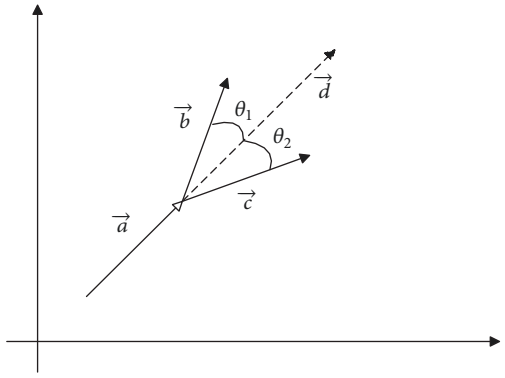


Figure 3: Relative offset angle.

Table 1: Representation of trajectory segment.

Trajectory section			
RT		FT	
x_1	y_1	roa_1	rl_1
x_2	y_2	roa_2	rl_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$
x_{m-1}	y_{m-1}	roa_{m-1}	rl_{m-1}

this paper. Unlike ordinary image processing, we deal with the feature matrix here and need to make some improvements to the universal network model. The network model structure and training process are shown in Figure 4. The CNN model in this paper is the improvement of the universal model of CNN introduced in literature [30], especially in the network layer architecture of convolutional layer and pooling layer. The detailed steps in the dotted box of Figure 4 are depicted in Figure 5.

The process of doing one network training is as follows:

(1) Relative trajectory (RT) and feature trajectory (FT) are jointly used to represent a trajectory segment; they are input into convolutional layer of CNN to extract trajectory features.

(2) Maximum pooling operations are used to reduce the amount of parameters after each convolutional operation; then all outputs are connected to a matrix for each feature metric.

(3) Fully connected layer is used to synthesize the features extracted from the front to obtain a 1×2 matrix.

(4) Softmax and cross entropy operations (Softmax operation calculates the probability value of classification results; cross entropy calculates the distance between CNN classification results and real classification, which is called loss) are executed on the matrix obtained by the fully connected layer to get loss. The total Softmax is defined as Softmax = αSoftmax1 + $(1 - \alpha)$Softmax2, where α is the weight.

(5) Convolutional layer and fully connected layer parameters W and b (W is a weight parameter and b is a bias

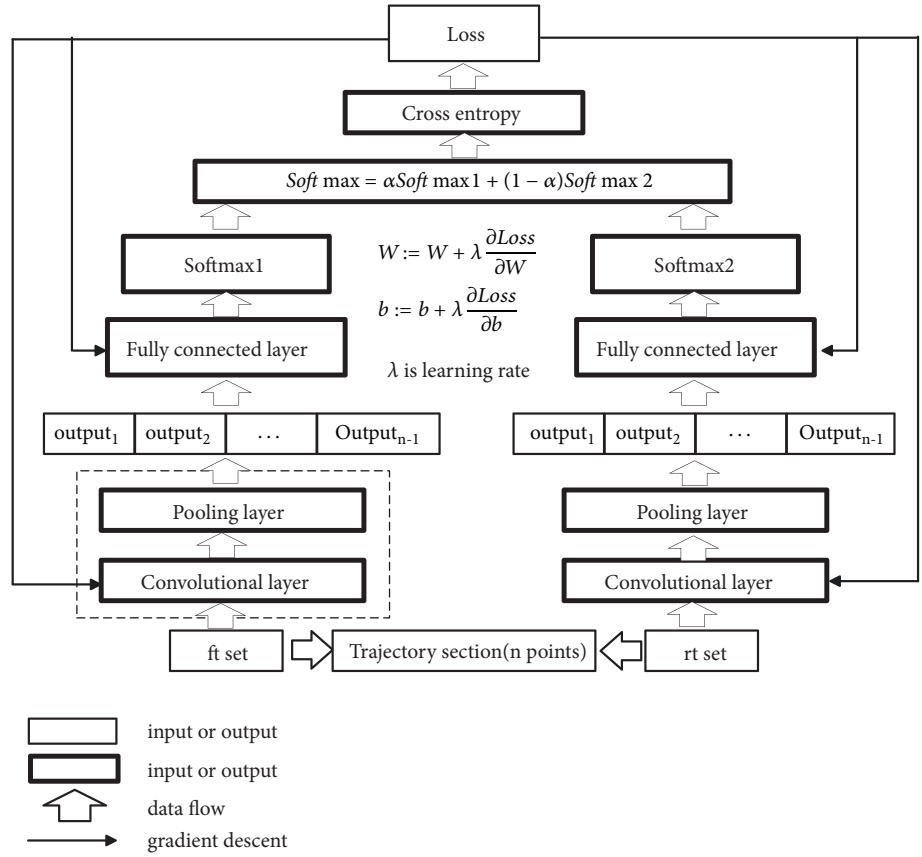


Figure 4: Training process for CNN.

item) are updated by gradient descent method based on loss equation.

FT and RT describe trajectories in different ways, and their dimensions are different, so they cannot be put together. The network structure on the left is FT feature detection network and the one on the right is RT feature detection network. For each iteration process, we send the corresponding RT and FT to networks separately. Then we use gradient descent method (Gradient descent is one of the iterative methods, which can be used to solve the least squares problem. The calculation of gradient descent method is to solve the minimum value along the direction of gradient descent.) to update the weight parameter W and bias parameter b of convolutional layer and fully connected layer. After updating the network model for a certain number of times, the model can distinguish the real trajectory from the dummy trajectories.

In these two layers, the following operations are executed:

(1) We arrange and transform feature trajectory into matrix in time sequence. Feature trajectory is represented as a $(n-1)\times 2$ matrix as illustrated in the right part of Table 1.

(2) We use n-1 types of filters to extract the arrangement characteristics of feature trajectory, and there are 128 filters in each type. For the first k type filter, we extract characteristics of adjacent k trajectory segments in turn by convolutional operation to obtain a $(n-k)\times 1$ matrix. Then $128n-128$ matrices that are output from convolutional operation are, respectively, executing the maximum pooling operation.

(3) After operation of a pooling layer, we obtain $n-1$ outputs and each output has 128 matrices. Then, we connect these $n-1$ outputs into a matrix and send them to the fully connected layer shown in Figure 4 for the next step.

3.4.2. Trajectory Detection Schemes. The network architecture of dummy trajectory detection is shown in Figure 6. Different from the CNN training process, the forward propagation process is only executed once to obtain Softmax; then the final judgment is made according to Softmax.

We first need to express the detected trajectory sections with FT and RT, respectively, and then put them into CNN to run in the direction indicated by the arrow once. Finally, we can obtain the detection result.

4. Experiments and Analysis

We use the trajectories from Microsoft research GeoLift project as the testing data set; 182 pieces of users' track data were collected from April 2007 to August 2012. These data

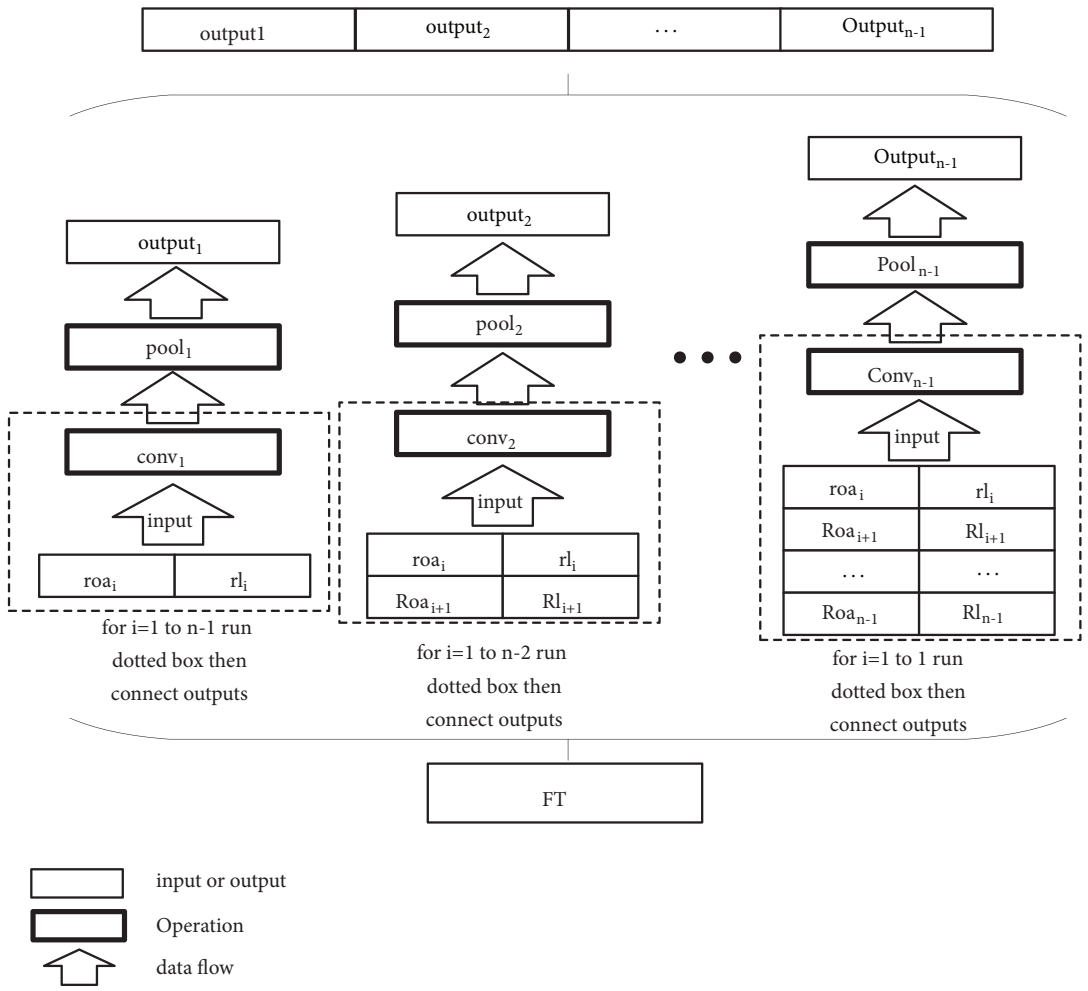


Figure 5: The convolutional layer and pooling layer of FT d network.

sets contain a series of time-ordered points, each containing latitude, longitude, elevation, and other pieces of information. In the experiments, we use 17621 tracks whose total distance is about 1200000 kilometers, and the total time is more than 48000 hours. These data not only record the user's location at home and at work but also track a wide range of outdoor activities, such as shopping, traveling, hiking, and cycling.

The experimental environment is as follows: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz, with 6G memory.

Programming language was Python.

We use 5s as a time interval to extract trajectory points and then divide these trajectories into sections and each one has 8 points. These sections are real trajectory section set. In addition, we extract and synthesize the algorithm fragment of the current widely used dummy trajectory generation algorithms to generate dummy trajectories. The main steps are shown in Algorithm 3.

We use Algorithm 3 to generate dummy trajectory section set and then put dummy trajectories and real trajectories into CNN for training.

In Table 2, the confusion matrix is listed, which is also known as error matrix. It is a standard format for accuracy evaluation, which is expressed in the form of matrix of N rows and N columns. In AI, confusion matrix is a visualization tool, especially for supervised learning. Unsupervised learning is generally called matching matrix.

Table 2: Confusion matrix.

Predicted value	True value	
	0 (dummy)	1 (real)
0 (dummy)	TN (True Negative)	FN (False Negative)
1 (real)	FP (False Positive)	TP (True Positive)

In our CNN-based dummy trajectory detection scheme, the weight parameter α of RT and FT has a great influence on the detection results. We take seven different values for α and train seven CNN networks. Then we use six different classifier evaluation indicators to evaluate CNN networks. The six evaluation indicators are as follows.

$$\text{Recognition rate} = \frac{TN}{TN + FP}$$

$$\text{Erroneous judgement rate} = \frac{FN}{FN + TP}$$

Input: real trajectory section set
Output: dummy trajectory section set
(1) Procedure
(2) **for** each real trajectory section in real trajectory section set **do**:
(3) **for** i=1 to k **do**:
(4) **for** j=1 to n-1**do**:
(5) randomly selected rotation angle in $(\pi/20, \pi/3) \cup (-\pi/3, -\pi/20)$
(6) Randomly selected expansion rates in(0.5, 1.5)
(7) rotate and expanse trajectory segment j
(8) **end for**
(9) **end for**
(10) **end for**

Algorithm 3: Dummy trajectory section generation.

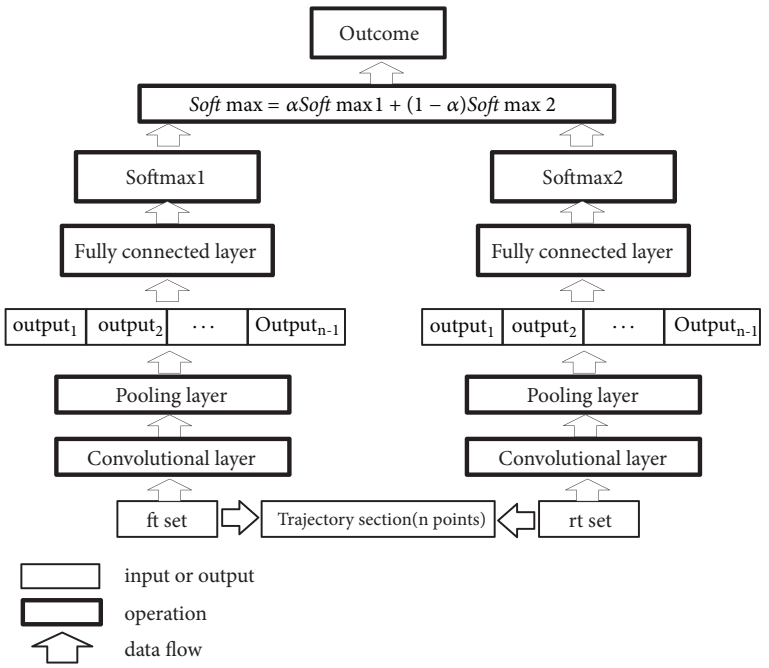


Figure 6: The network architecture of dummy trajectory detection.

$$\begin{aligned} \text{Precision rate P} &= \frac{TP}{TP + FP} \\ \textit{Recall rate } R &= \frac{TP}{TP + FN} \\ \text{F1} - \text{Measure} &= \frac{2 \times P \times R}{P + R} \\ \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{1}$$

Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. That is to say, when the loss function is 0-1 loss, accuracy is the accuracy of the test data set.

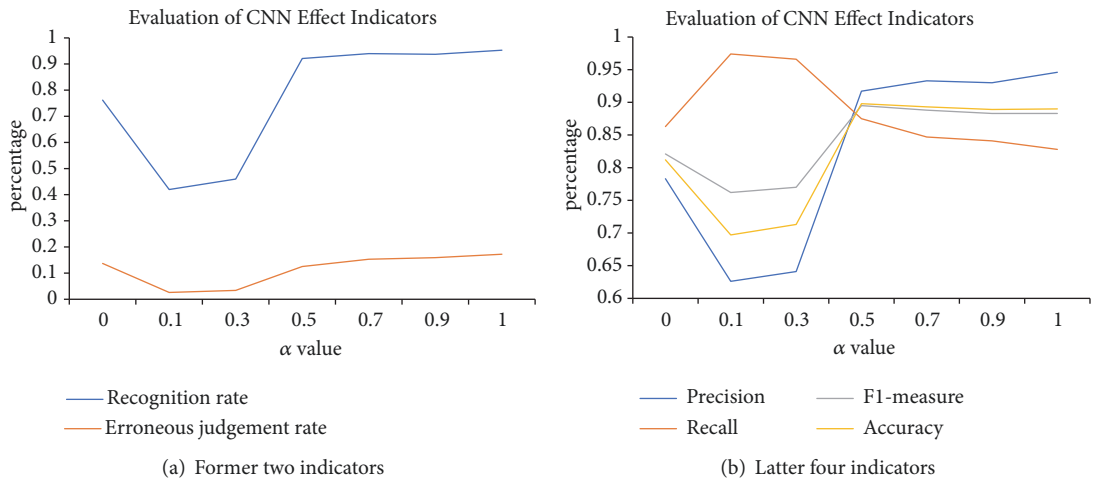
Precision calculates the proportion of items that "should be retrieved" among all items retrieved.

The recall rate calculates the proportion of all items retrieved to all items that should be retrieved.

The comprehensive evaluation index (F1-Measure) is the weighted harmonic average of Precision and Recall. P and R indicators are sometimes contradictory, considering both precision and recall. It is easy to understand that F1 combines the results of P and R. When F1 is higher, the experimental method is more ideal.

For different α values, the values of the six evaluation indicators are shown in Figure 7.

From Figure 7, it can be concluded that the FT attribute of the trajectory section controls the recognition rate; the higher the weight of FT attribute is, the higher the recognition rate is. And the RT attribute of the trajectory section controls the erroneous judgement rate; the higher the weight of RT attribute is, the lower the erroneous judgement rate is. F1-measure and the accuracy are comprehensive global



(a) Former two indicators (b) Latter four indicators

Figure 7: Evaluation indicators of different α value.

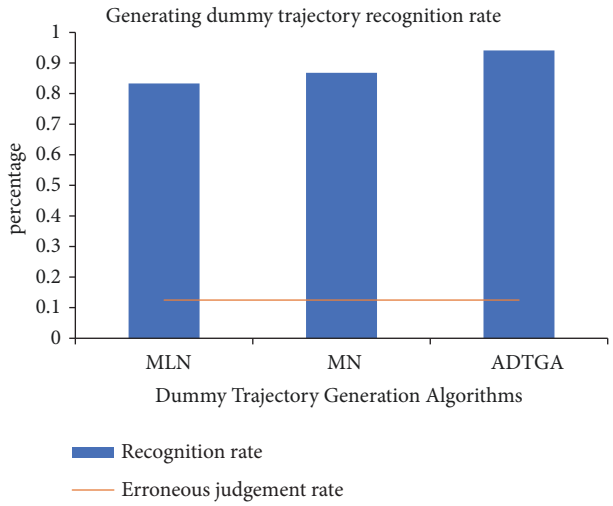


Figure 8: Detection results of dummy trajectory generation algorithms with $\alpha = 0.5$.

evaluation indicators, and the maximum value is obtained when α is 0.5. At the same time, when α is 0.5, the recognition rate is high and the erroneous judgement is low. Therefore, we choose the weight parameter $\alpha = 0.5$.

To verify the efficiency of CNN detection network, three classical algorithms are used to generate dummy trajectories, in which MLN and MN algorithms are proposed in [18], and ADTGA algorithm is proposed by [15]. The detection results are shown in Figure 8.

We use a data set named GeoLife GPS Trajectories as our real trajectory data set and use MLN, MN, and ADTGA to generate large number of dummy trajectories and then randomly select 1000 dummy trajectory sections from the dummy trajectory set and put them into the trained CNN detection network. For MLN, the dummy trajectory recognition rate is 83.3%. For MN, the dummy trajectory recognition rate is 86.8%. And, for ADTGA, the dummy trajectory recognition rate is 94.1%. But the erroneous judgement rate of real trajectories is only 12.5%.

Many dummy trajectory algorithms are improved by the three algorithms mentioned above, for example, literatures [13, 24]. The most significant improvement is the selection of dummy trajectories rather than the generation of dummy trajectory. Therefore, the improved dummy trajectory generation algorithm cannot reduce the detection rate. The experiment illustrates that our dummy trajectory detection scheme based on CNN can detect the dummy trajectory with high recognition rate, while keeping the low erroneous judgement rate. Generally, a complete trajectory has multiple trajectory sections. As shown in Figure 9, for dummy trajectories with multiple trajectory sections, we calculate their recognition rate and the erroneous judgement rate of real trajectories.

With the increase of the number of trajectory sections, the trend of the recognition rate is also increasing, while the erroneous judgement rate shows a downward trend, and the recognition rate is above 90%, and the erroneous rate is below 10%.

5. Conclusion

We have studied many algorithms to generate the dummy trajectories to protect privacy, most of which only take into account the geometric meaning of trajectories without considering the human mobility model. In order to address this weakness, we define two trajectory representation methods and put these two trajectory representation methods of real and dummy trajectories into the improved CNN for training. Experiments show that the deep learning machine CNN is universal; it can identify more than 90% of dummy trajectories that are generated using the current mainly algorithm; meanwhile its erroneous judgement rate is below 10%.

Indeed, our detection scheme cannot be applied to all dummy trajectory generation algorithms. There are two kinds of dummy trajectory generation algorithms; our detection

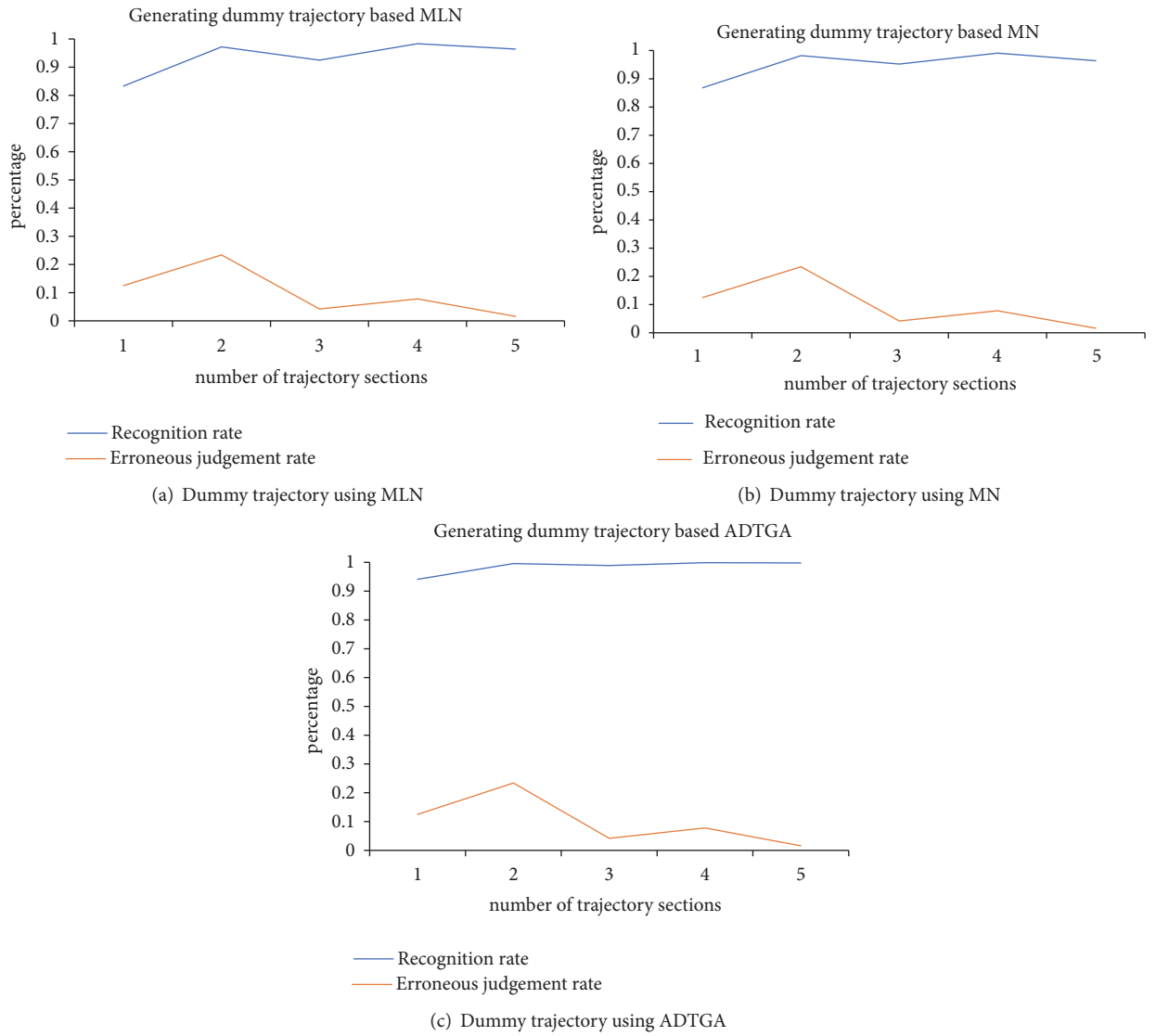


Figure 9: Recognition rate and erroneous rate under the assumption that the discrimination is 0.5.

scheme is powerless. One is simple rotation algorithm and the other is to select similar trajectories or historical trajectories of other users as dummy trajectories. However, the first dummy trajectory generation algorithm is not flexible enough to meet the real background, and the second one needs to collect a large amount of historical trajectory information of the surrounding users, which is very difficult. So these two methods of dummy trajectory generation are difficult to be used in practice. Generally speaking, our dummy trajectory detection scheme has a high detection rate for the dummy trajectory generated by the deformed dummy trajectory generation method.

Our experiments show that the common flaws of the dummy trajectory generation algorithms up till now are that they only consider the geometry of the trajectory points and the trajectory segments and regard them as the products isolated from human behavior and the products of random probability. It is debatable whether such a convolutional neural network learning machine can act as a filter to filter out most of the dummy trajectories which do not conform to reality, as well as leaving behind some dummy trajectories which mix the spurious with the genuine. After all, in the era of exploding CPU and GPU performance, the time cost of generating redundant dummy trajectory sets is negligible.

Data Availability

The data used to support the findings of the manuscript are obtained from Microsoft research GeoLift project; they can be used freely and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[2] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[3] M. Tang, Q. Wu, G. Zhang, L. He, and H. Zhang, "A new scheme of LBS privacy protection," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '09)*, pp. 1–6, Beijing, China, September 2009.

[4] W. He, "Research on LBS privacy protection technology in mobile social networks," in *Proceedings of the 2nd IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC '17)*, pp. 73–76, Chongqing, China, March 2017.

[5] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 348–357, ACM, New York, NY, USA, November 2009.

[6] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 547–555, IEEE, Phoenix, AZ, USA, April 2008.

[7] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and B. Xu, "L2P2: location-aware location privacy protection for location-based services," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 1996–2004, IEEE, Orlando, FL, USA, March 2012.

[8] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[9] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[10] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[11] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and K-anonymity," *World Wide Web*, pp. 1–11, 2018.

[12] Y. Liu, Y. Wang, X. Wang, Z. Xia, and J. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Computer Networks*, vol. 148, pp. 340–348, 2019.

[13] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *Proceedings of the 8th International Conference on Mobile Data Management (MDM '07)*, pp. 278–282, Mannheim, Germany, May 2007.

[14] P.-R. Lei, W.-C. Peng, I.-J. Su, and C.-P. Chang, "Dummy-based schemes for protecting movement trajectories," *Journal of Information Science and Engineering*, vol. 28, no. 2, pp. 335–350, 2012.

[15] X. Wu and G. Sun, "A novel dummy-based mechanism to protect privacy on trajectories," in *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW '14)*, pp. 1120–1125, Shenzhen, China, December 2014.

[16] R. Kato, M. Iwata, T. Hara et al., "A dummy-based anonymization method based on user trajectory with pauses," in *Proceedings of the 20th ACM International Conference on Advances in Geographic Information Systems (AGIS '12)*, pp. 249–258, ACM, New York, NY, USA, 2012.

[17] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM '08)*, pp. 65–72, Beijing, China, April 2008.

[18] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, Santorini, Greece, July 2005.

[19] H. Ngo and J. Kim, "Location privacy via differential private perturbation of cloaking area," in *Proceedings of the IEEE 28th Computer Security Foundations Symposium (CSF '15)*, pp. 63–74, Verona, Italy, July 2015.

[20] L. Ou, Z. Qin, Y. Liu, H. Yin, Y. Hu, and H. Chen, "Multi-user location correlation protection with differential privacy," in *Proceedings of the IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS '16)*, pp. 422–429, Wuhan, China, December 2016.

[21] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proceedings of the 21st International Conference on Data Engineering Workshops (ICDEW '05)*, p. 1248, Tokyo, Japan, April 2005.

[22] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '14)*, pp. 754–762, IEEE, Toronto, Canada, April-May 2014.

[23] T. Hara, Y. Arase, A. Yamamoto, X. Xie, M. Iwata, and S. Nishio, "Location anonymization using real car trace data for location based services," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication (ICUIMC '14)*, pp. 1–8, ACM, New York, NY, USA, January 2014.

[24] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 16–23, ACM, New York, NY, USA, June 2008.

[25] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio, "A user location anonymization method for location based services in a real environment," in *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '10)*, pp. 398–401, ACM, November 2010.

[26] R. Kato, M. Iwata, T. Hara, Y. Arase, X. Xie, and S. Nishio, "User location anonymization method for wide distribution of dummies," in *Database and Expert Systems Applications*, vol. 8056 of *Lecture Notes in Computer Science*, pp. 259–273, Springer, Berlin Heidelberg, 2013.

[27] R. Lan, J. He, S. Wang, Y. Liu, and X. Luo, "A parameter-selection-based chaotic system," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 3, pp. 492–496, 2019.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[29] L. Bottou, Y. Bengio, and Y. Le Cun, "Global training of document processing systems using graph transformer networks,"

in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 489–494, San Juan, PR, USA, June 1997.

[30] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*, pp. 1746–1751, Doha, Qatar, October 2014.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 1348639, 15 pages
https://doi.org/10.1155/2019/1348639

Research Article

Measuring the Sum-of-Squares Indicator of Boolean Functions in Encryption Algorithm for Internet of Things

Yu Zhou,[1] **Yongzhuang Wei**,[2] **and Fengrong Zhang**[3]

[1]*Science and Technology on Communication Security Laboratory, Chengdu 610041, China*
[2]*Department of Communication and Information Engineering, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*China University of Mining Technology School of Computer Science and Technology, Xuzhou 221116, China*

Correspondence should be addressed to Yu Zhou; zhouyu.zhy@tom.com

Received 24 January 2019; Accepted 27 February 2019; Published 27 March 2019

Guest Editor: Fagen Li

Encryption algorithm has an important application in ensuring the security of the Internet of Things. Boolean function is the basic component of symmetric encryption algorithm, and its many cryptographic properties are important indicators to measure the security of cryptographic algorithm. This paper focuses on the sum-of-squares indicator of Boolean function; an upper bound and a lower bound of the sum-of-squares on Boolean functions are obtained by the decomposition Boolean functions; some properties and a search algorithm of Boolean functions with the same autocorrelation (or cross-correlation) distribution are given. Finally, a construction method to obtain a balanced Boolean function with small sum-of-squares indicator is derived by decomposition Boolean functions. Compared with the known balanced Boolean functions, the constructed functions have the higher nonlinearity and the better global avalanche characteristics property.

1. Introduction

The Internet of Things is an important part of the new generation of information technology and also an important stage of information development. But the Internet of Things is being threatened and attacked by more and more potential threats and attacks [1, 2]. As Internet of Things evolves, these networks, and many others, will be connected with security and management capabilities and so forth.

The current-time wireless sensor network is attacked by hackers from time to time, and it has put up a new challenge for information security. With wireless communication, low cost, resource constraints, and so forth, current threats include differential power analysis, kinds of keys decryption, Trojan attacks, virus damage, and physical method.

Because of the special use of wireless sensor, the design of key storage, distribution, and encryption or decryption algorithm is inconvenient. Therefore, we need to be practical and convenient for the cryptographic algorithm in resource-constrained environment, the most basic of which is to clarify the cryptographic properties of cryptographic components.

The scenarios used in the Internet of Things are mostly resource-constrained, so its cryptographic algorithm requires some hardware and software requirements, low power consumption, moderate security intensity, and limited resource area, which makes the design of such cryptographic algorithm more difficult. Therefore, the research of cryptographic components in cryptographic algorithm is very important.

Symmetric cryptographic algorithm is the most widely used in cooperative networks. Its advantage is to ensure the confidentiality of communication data. If the algorithm is authenticated, it can ensure the integrity of communication data. Block cipher and stream cipher are two main design directions. Boolean function, as the most basic and widely used cryptographic component, has been highly studied by scholars, for example, linear feedback shift registers (LFSR), S-box, and MDS.

Boolean functions have many cryptographical indicators, including balance, high nonlinearity, high algebraic degree, resilience, propagation characteristic [3], global avalanche

characteristic (*GAC*) [4], algebraic immunity [5], and transparency order [6]. Among these properties, *GAC* can link with other cryptographic indicators. In 1995, Zhang and Zheng introduced the global avalanche characteristic (*GAC* [4]: the sum-of-squares indicator (σ_f), the absolute indicator ($\triangle_f$)) for an n-variable Boolean function $f(x)$, and they gave the lower and the upper bounds on the two indicators. Reference [4] implied that the smaller σ_f and $\triangle_f$, the better the *GAC* of a Boolean function. In 1998, Son et al. [7] gave a lower bound on these indicators for a balanced Boolean function: $\sigma_f \geq 2^{2n} + 2^{n+3}$ and $\triangle_f \geq 8 (n \geq 3)$. Sung et al. [8] improved these results and provided a bound on the sum-of-squares indicator of balanced functions satisfying the propagation criterion with respect to t vectors. In 2010, [9] generalized the *GAC* and put up a new criterion based on the cross-correlation functions: the sum-of-squares indicator ($\sigma_{f,g}$) and the absolute indicator ($\triangle_{f,g}$) for two n-variable Boolean functions $f(x), g(x)$; they gave the lower and the upper bounds on the two indicators. Reference [10] derived a new bound on the sum-of-squares indicator and gave a method to construct balanced Boolean functions with $n (n \geq 6)$ variables by the disjoint spectra functions, where n is an even integer, satisfying strict avalanche criterion, high nonlinearity, and lower *GAC*.

Meanwhile, some authors gave lots of constructions of Boolean functions with good *GAC*, Tang [11] gave a method to construct balanced Boolean functions of n variables, the constructed functions possess the highest nonlinearity and the better global avalanche characteristics (*GAC*) property, but they only obtained an upper bound of *GAC*. Reference [12] gave a method to construct high nonlinearity Boolean function. These constructions had not considered Boolean functions with the same autocorrelation distributions or the same cross-correlation distributions. If these functions have the same autocorrelation distributions or the same cross-correlation distributions, then these Boolean functions have the same *GAC* [4], the same transparency order [6], the same nonlinearity, the same absolute value of Walsh spectrum, the same correlation immunity, the same propagation criterion, and so forth. Thus, this paper will construct a Boolean function with small *GAC*, and we give some relationships of the sum-of-squares indicator between an n-variable Boolean function and four $(n-2)$–variable decomposition Boolean functions; the relationships are based on construction on Boolean functions with good global avalanche characteristics.

Based on the above consideration, we study the following questions:

(1) What is a clear characterization of four $(n-2)$-variable decomposition functions, if the sum-of-squares indicator of an n-variable Boolean function is lower? This study provides theoretical support for the security of lightweight dynamic cryptographic algorithms in the Internet of Things.

(2) What are the cross-correlation properties of any two Boolean functions, if Boolean functions have the same autocorrelation distribution? This research lays a foundation for lightweight dynamic cryptographic algorithms in the Internet of Things.

(3) How to construct a Boolean function with good global avalanche characteristics. This study provides some algorithm component for the lightweight dynamic cryptographic algorithms in the Internet of Things.

The rest of this paper is organized as follows: Section 2 introduces some basic definitions. In Section 3, an upper bound on the sum-of-squares indicator of n-variable Boolean function by using four decomposition $(n-2)$-variable Boolean functions is given. Section 4 gives some properties of a Boolean function with the upper bound on the sum-of-squares indicator. In Section 5, we give a construction of one Boolean function with small sum-of-squares indicator by the disjoint spectrum method. Finally, Section 6 concludes this paper.

2. Preliminaries

Let $\mathbb{B}_n$ denote the set of n variables Boolean functions. We denote by $\oplus$ the additions in $\mathbb{F}_2$, in $\mathbb{F}_2^n$, and in $\mathbb{B}_n$. Every Boolean function $f(x) \in \mathbb{B}_n$ admits a unique representation called its algebraic normal form (*ANF*) as a polynomial over $\mathbb{F}_2$:

$$f(x_1, \ldots, x_n) = a_0 \oplus \bigoplus_{1 \leq i \leq n} a_i x_i \oplus \bigoplus_{1 \leq i < j \leq n} a_{i,j} x_i x_j \oplus \cdots \oplus a_{1,\ldots,n} x_1 x_2 \cdots x_n \tag{1}$$

where the coefficients $a_0, a_i, a_{i,j}, \ldots, a_{1,\ldots,n} \in \mathbb{F}_2$. The algebraic degree, $\deg(f)$, is the number of variables in the highest-order term with nonzero coefficient. The support of a Boolean function $f(x) \in \mathbb{B}_n$ is defined as $\mathrm{Supp}(f) = \{(x_1, \ldots, x_n) \mid f(x_1, \ldots, x_n) = 1\}$. We say that a Boolean function $f(x)$ is balanced if its truth table contains an equal number of ones and zeros, that is, if its Hamming weight equals 2^{n-1}. A Boolean function is affine if there exists no term of degree > 1 in the *ANF* and the set of all affine functions is denoted by $\mathbb{A}_n$. An affine function with constant term equal to zero is called a linear function.

Definition 1. The Walsh spectrum of $f(x) \in \mathbb{B}_n$ is defined as

$$\mathscr{F}(f \oplus \varphi_\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \alpha x}, \tag{2}$$

where $\varphi_\alpha = \alpha x = \alpha_1 x_1 \oplus \alpha_2 x_2 \oplus \cdots \oplus \alpha_n x_n$.

Definition 2. The cross-correlation function between $f(x), g(x) \in \mathbb{B}_n$ is defined as

$$\triangle_{f,g}(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus g(x \oplus \alpha)}, \quad \alpha \in \mathbb{F}_2^n. \tag{3}$$

If $f(x) = g(x)$, then $\triangle_f(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(x \oplus \alpha)}$.

Two n-variable Boolean functions $f(x), g(x)$ are called to be perfectly uncorrelated if $\triangle_{f,g}(\alpha) = 0$ for all $\alpha \in \mathbb{F}_2^n$ and are called to be uncorrelated of degree k if $\triangle_{f,g}(\alpha) = 0$ for all $\alpha \in \mathbb{F}_2^n$ such that $0 \leq wt(\alpha) \leq k$.

Definition 3 (see [9]). Let $f(x), g(x) \in \mathbb{B}_n$; the sum-of-squares indicator of the cross-correlation between $f(x)$ and $g(x)$ is defined as

$$\sigma_{f,g} = \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,g}^2(\alpha); \tag{4}$$

the absolute indicator of the cross-correlation between $f(x)$ and $g(x)$ is defined as

$$\triangle_{f,g} = \max_{\alpha \in \mathbb{F}_2^n} \left| \triangle_{f,g}(\alpha) \right|. \tag{5}$$

The above indicators are called the global avalanche characteristics between two Boolean functions. If $f(x) = g(x)$, then

$$\begin{aligned} \sigma_f &= \sum_{\alpha \in \mathbb{F}_2^n} \triangle_f^2(\alpha), \\ \triangle_f &= \max_{\alpha \in \mathbb{F}_2^n, wt(\alpha) \neq \mathbf{0}^n} \left| \triangle_f(\alpha) \right|, \end{aligned} \tag{6}$$

and the two indicators are the global avalanche characteristics of Boolean functions (*GAC* [4]).

In order to study cross-correlation distributions between any two Boolean functions, we need the following definition.

Definition 4 (see [13]). Let $f(x), g(x) \in \mathbb{B}_n$. If $D_a(f,g) : x \longmapsto f(x) \oplus g(x \oplus a)$ is constant, a is said to be a *linear structure* of f and g. For convenience, let

$$\begin{aligned} U_{f,g}^0 &= \{a \in \mathbb{F}_2^n \mid f(x) \oplus g(x \oplus a) = 0, \ \forall x \in \mathbb{F}_2^n\}; \\ U_{f,g}^1 &= \{a \in \mathbb{F}_2^n \mid f(x) \oplus g(x \oplus a) = 1, \ \forall x \in \mathbb{F}_2^n\}; \end{aligned} \tag{7}$$

if $\mathbf{0}^n \in U_{f,g}$, it is easy to know that $U_{f,g}^0$ and $U_{f,g} = U_{f,g}^0 \cup U_{f,g}^1$ are linear subspaces of $\mathbb{F}_2^n$.

In Definition 4, if $f(x) = g(x)$, then $U_f^0 = \{a \in \mathbb{F}_2^n \mid f(x) \oplus f(x \oplus a) = 0, \ \forall x \in \mathbb{F}_2^n\}$; $U_f^1 = \{a \in \mathbb{F}_2^n \mid f(x) \oplus f(x \oplus a) = 1, \ \forall x \in \mathbb{F}_2^n\}$. U_f^0 and $U_f = U_f^0 \cup U_f^1$ are linear subspaces of $\mathbb{F}_2^n$.

In [13], the authors obtained some properties of any two Boolean functions with the same autocorrelation distribution; let $T_n(f)$ and $T_{n\times n}(f,g)$ be functions set with the same autocorrelation distribution and the cross-correlation distribution of given $f(x), g(x) \in \mathbb{B}_n$, respectively:

$$\begin{aligned} T_n(f) = \big\{ & g(x) \in \mathbb{B}_n \mid \triangle_g(\alpha_i) = \triangle_f(\alpha_i), \ \forall \alpha_i \\ & \in \mathbb{F}_2^n \ (0 \le i \le 2^n - 1), \ f(x) \in \mathbb{B}_n \big\}. \end{aligned} \tag{8}$$

And

$$\begin{aligned} T_{n\times n}(f,g) = \big\{ & (r(x), t(x)) \in \mathbb{B}_n \times \mathbb{B}_n \mid (\triangle_{r,t}(\alpha) \\ & = \triangle_{f,g}(\alpha), \ \forall \alpha_i \\ & \in \mathbb{F}_2^n \ (0 \le i \le 2^n - 1), \ f(x), g(x) \in \mathbb{B}_n \big\}. \end{aligned} \tag{9}$$

We denote a Boolean function $f(x) \in \mathbb{B}_n$ by $\overline{f} = f_0 \times 2^0 + f_1 \times 2^1 + \cdots + f_{2^n-1} \times (2^{2^n-1})$. For example, taking $n = 3$, the Boolean function with truth table $(f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7) = (1,0,0,0,1,1,0,1)$ is written as $\overline{f} = 177$.

Denote $\mathbf{0}^n = (0, 0, \ldots, 0) \in \mathbb{F}_2^n$ in this paper.

3. The Upper Bound on the Sum-of-Squares between an n-Variables Boolean Function and $(n-2)$-Variable Decomposition Functions

In this section, we give an expression for the sum-of-squares indicator of an n-variable Boolean function. This result is important to the following sections.

In order to give the relationship of the sum-of-squares indicator between one Boolean function and four decomposition Boolean functions, we need the following Lemma 5.

Lemma 5 (see [13]). *Let $h(x), g(x) \in \mathbb{B}_n$. Then*

$$\sum_{\alpha \in \mathbb{F}_2^n} \triangle_h(\alpha) \triangle_g(\alpha) = \sum_{e \in \mathbb{F}_2^n} \triangle_{h,g}^2(e) = \sigma_{h,g}. \tag{10}$$

Lemma 5 gave a relationship between autocorrelation functions and cross-correlation functions. Reference [14] gives the relationship between the sum-of-squares indicator on an n-variable Boolean function and four decomposition $(n-2)$-variable Boolean functions in the following.

Lemma 6 (see [14]). *Let $f(x_n, x_{n-1}, x) = (x_n \oplus 1)(x_{n-1} \oplus 1) f_1(x) \oplus (x_n \oplus 1) x_{n-1} f_2(x) \oplus x_n (x_{n-1} \oplus 1) f_3(x) \oplus x_n x_{n-1} f_4(x)$; $x_n, x_{n-1} \in \mathbb{F}_2$, $x \in \mathbb{F}_2^{n-2}$. Then*

$$\begin{aligned} \sigma_f &= \sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4} \\ &\quad + 6\left[\sigma_{f_1,f_2} + \sigma_{f_3,f_4} + \sigma_{f_1,f_3} + \sigma_{f_2,f_4} + \sigma_{f_1,f_4} + \sigma_{f_2,f_3}\right] \\ &\quad + 8 \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}(\alpha) \triangle_{f_3,f_4}(\alpha) \\ &\quad + 8 \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}(\alpha) \triangle_{f_2,f_4}(\alpha) \\ &\quad + 8 \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_4}(\alpha) \triangle_{f_2,f_3}(\alpha). \end{aligned} \tag{11}$$

Based on Lemmas 5 and 6, we have the upper bound on σ_f for any n-variable Boolean function $f(x) \in \mathbb{B}_n$.

Theorem 7. *Let $f(x_n, x_{n-1}, x) = (x_n \oplus 1)(x_{n-1} \oplus 1) f_1(x) \oplus (x_n \oplus 1) x_{n-1} f_2(x) \oplus x_n (x_{n-1} \oplus 1) f_3(x) \oplus x_n x_{n-1} f_4(x)$; $x_n, x_{n-1} \in \mathbb{F}_2$, $x \in \mathbb{F}_2^{n-2}$. Then*

$$\sigma_f \le \sum_{1 \le i \le 4} \sigma_{f_i} + 6 \sum_{1 \le i < j \le 4} \sigma_{f_i,f_j}$$

$$+ 8\left[\sqrt{\sigma_{f_1,f_2}\sigma_{f_3,f_4}} + \sqrt{\sigma_{f_1,f_3}\sigma_{f_2,f_4}} + \sqrt{\sigma_{f_1,f_4}\sigma_{f_2,f_3}}\right], \tag{12}$$

with the equality holding if and only if $\triangle_{f_1,f_2}(\alpha) = \triangle_{f_3,f_4}(\alpha)$, $\triangle_{f_1,f_3}(\alpha) = \triangle_{f_2,f_4}(\alpha)$, *and* $\triangle_{f_1,f_4}(\alpha) = \triangle_{f_2,f_3}(\alpha)$ *for any* $\alpha \in \mathbb{F}_2^{n-2}$.

Proof. Note that, for any a_i, b_i, Cauchy inequality holds:

$$\sum_{i=1}^{n} a_i b_i \le \left(\sum_{i=1}^{n} a_i^2\right)^{1/2} \left(\sum_{i=1}^{n} b_i^2\right)^{1/2}, \tag{13}$$

with equality holding if and only if $a_i = b_i$ for any i $(1 \le i \le n)$.

Thus, based on Definition 3, we have

$$\begin{aligned}
&\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha) \\
&\le \left(\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}^2(\alpha) \sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_3,f_4}^2(\alpha)\right)^{1/2} \\
&= \sqrt{\sigma_{f_1,f_2}\sigma_{f_3,f_4}}, \\
&\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}(\alpha)\triangle_{f_2,f_4}(\alpha) \\
&\le \left(\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}^2(\alpha) \sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_2,f_4}^2(\alpha)\right)^{1/2} \\
&= \sqrt{\sigma_{f_1,f_3}\sigma_{f_2,f_4}}, \\
&\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_4}(\alpha)\triangle_{f_2,f_3}(\alpha) \\
&\le \left(\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_4}^2(\alpha) \sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_2,f_3}^2(\alpha)\right)^{1/2} \\
&= \sqrt{\sigma_{f_1,f_4}\sigma_{f_2,f_3}}.
\end{aligned} \tag{14}$$

This result is proven. □

Reference [15] gave the relationship between $\sigma_{f,g}$ and σ_f, σ_g for any Boolean function $f, g \in \mathbb{B}_n$.

Lemma 8 (see [15]). *Let* $f(x), g(x) \in \mathbb{B}_n$. *Then* $\sigma_{f,g} \le \sqrt{\sigma_f\sigma_g}$; *the equality holds if and only if* $|\mathcal{F}(f \oplus \varphi_\alpha)| = |\mathcal{F}(g \oplus \varphi_\alpha)|$ *for all* $\alpha \in \mathbb{F}_2^n$ *or if and only if* $\triangle_f(\alpha) = \triangle_g(\alpha)$ *for all* $\alpha \in \mathbb{F}_2^n$.

Furthermore, according to Lemma 8 and Theorem 7, we have the following theorem.

Theorem 9. *Let* $f(x_n, x_{n-1}, x) = (x_n \oplus 1)(x_{n-1} \oplus 1)f_1(x) \oplus (x_n \oplus 1)x_{n-1}f_2(x) \oplus x_n(x_{n-1} \oplus 1)f_3(x) \oplus x_n x_{n-1} f_4(x)$; $x_n, x_{n-1} \in \mathbb{F}_2$, $x \in \mathbb{F}_2^{n-2}$. *Then*

$$\begin{aligned}
2^{2n} &\le \sigma_f \\
&\le \sum_{1\le i\le 4} \sigma_{f_i} + 6 \sum_{1\le i<j\le 4} \sqrt{\sigma_{f_i}\sigma_{f_j}} \\
&\quad + 24\left(\sigma_{f_1}\sigma_{f_2}\sigma_{f_3}\sigma_{f_4}\right)^{1/4},
\end{aligned} \tag{15}$$

and, furthermore, we have the following:

(1) The right equality holds if and only if the two following conditions are satisfied:

(⋆) $\triangle_{f_1}(\alpha) = \triangle_{f_2}(\alpha) = \triangle_{f_3}(\alpha) = \triangle_{f_4}(\alpha)$ *for all* $\alpha \in \mathbb{F}_2^{n-2}$;

(⋆⋆) $\triangle_{f_1,f_2}(\alpha) = \triangle_{f_3,f_4}(\alpha)$, $\triangle_{f_1,f_3}(\alpha) = \triangle_{f_2,f_4}(\alpha)$, *and* $\triangle_{f_1,f_4}(\alpha) = \triangle_{f_2,f_3}(\alpha)$ *for any* $\alpha \in \mathbb{F}_2^{n-2}$.

(2) The left equality holds if and only if f *is a bent function.*

Remark 10. By Theorem 9, we know that $\sigma_f = \sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4}$ if and only if f_i and f_j $(1 \le i \ne j \le 4)$ are perfectly uncorrelated functions. The lower bound is easy reached; it is because we can find that f_i and f_j $(1 \le i \ne j \le 4)$ are perfectly uncorrelated functions. For example, let $f(x) = f(x_1, x_2, x_3, x_4) \in \mathbb{B}_4$ be a bent function and

$$\begin{aligned}
f_1(x, x_5, x_6, x_7, x_8) &= f(x) \oplus x_5, \\
f_2(x, x_5, x_6, x_7, x_8) &= f(x) \oplus x_6, \\
f_3(x, x_5, x_6, x_7, x_8) &= f(x) \oplus x_7, \\
f_4(x, x_5, x_6, x_7, x_8) &= f(x) \oplus x_8,
\end{aligned} \tag{16}$$

and then any two Boolean functions among f_1, f_2, f_3, f_4 are perfectly uncorrelated functions, and for every function $f_i(x)$ $(i = 1, 2, 3, 4)$ we have

$$\mathcal{F}(f_i \oplus \varphi_\alpha) = \begin{cases} 0, & 240 \text{ times}; \\ \pm 2^6, & 16 \text{ times}. \end{cases} \tag{17}$$

Thus, $\sigma_{f_i} = 2^{20}$. If $F(x, x_5, x_6, x_7, x_8, x_9, x_{10}) = (x_9 \oplus 1)(x_{10} \oplus 1)f_1 \oplus (x_9 \oplus 1)x_{10}f_2 \oplus x_9(x_{10} \oplus 1)f_3 \oplus x_9 x_{10} f_4$, then $\sigma_F = \sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4} = 4 \times 2^{20} = 2^{24}$; the lower bound can be reached.

Summary 1. Theorem 9 provides theoretical support for encryption algorithm, especially for the security of lightweight dynamic cryptographic algorithms in the Internet of Things [2].

4. Some Properties of Conditions (⋆) and (⋆⋆)

Theorem 9 induces an important problem: does there exist a (f_1, f_2, f_3, f_4)-pair satisfying conditions $(\star)$ and $(\star\star)$? We will analyze this question. We need the following lemma.

Lemma 11. *Let* $f(x), g(x), h(x) \in \mathbb{B}_n$.

(1) For any $\alpha \in \mathbb{F}_2^n$, $\triangle_f(\alpha) = \triangle_{f,g}(\alpha)$ if and only if $f(x) = g(x)$.

(2) For any $\alpha \in \mathbb{F}_2^n$, $\triangle_f(\alpha) = \triangle_{g,h}(\alpha)$; then $g(x) = h(x)$; furthermore, $\triangle_f(\alpha) = \triangle_g(\alpha) = \triangle_h(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$.

(3) For any $\alpha \in \mathbb{F}_2^n$, $\triangle_{f,g}(\alpha) = \triangle_{f,h}(\alpha)$ and $\triangle_f(\alpha) = \triangle_g(\alpha) = \triangle_h(\alpha)$; then $g(x) = h(x)$.

Proof. (1) According to the relationship between the cross-correlation function and the Walsh spectrum for $f(x), g(x) \in \mathbb{B}_n$, for any $\alpha \in \mathbb{F}_2^n$, we have

$$\triangle_{f,g}(\alpha) = \frac{1}{2^n} \sum_{\omega \in \mathbb{F}_2^n} (-1)^{\omega \cdot \alpha} \mathcal{F}(f \oplus \varphi_\omega) \mathcal{F}(g \oplus \varphi_\omega). \quad (18)$$

On one hand, since $\triangle_f(\alpha) = \triangle_{f,g}(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$, if $\alpha = \mathbf{0}^n$, we have

$$\begin{aligned} 2^n &= \triangle_f(\mathbf{0}^n) = \triangle_{f,g}(\mathbf{0}^n) \\ &= \frac{1}{2^n} \sum_{\omega \in \mathbb{F}_2^n} \mathcal{F}(f \oplus \varphi_\omega) \mathcal{F}(g \oplus \varphi_\omega) \end{aligned} \quad (19)$$

Thus

$$\sum_{\omega \in \mathbb{F}_2^n} \mathcal{F}(f \oplus \varphi_\omega) \mathcal{F}(g \oplus \varphi_\omega) = 2^{2n}. \quad (20)$$

Finally, by Parseval equality and (20), we have

$$\begin{aligned} \sum_{\omega \in \mathbb{F}_2^n} \left[\mathcal{F}(f \oplus \varphi_\omega) - \mathcal{F}(g \oplus \varphi_\omega)\right]^2 &= \sum_{\omega \in \mathbb{F}_2^n} \mathcal{F}^2(f \oplus \varphi_\omega) \\ &+ \sum_{\omega \in \mathbb{F}_2^n} \mathcal{F}^2(g \oplus \varphi_\omega) \\ &= -2 \sum_{\omega \in \mathbb{F}_2^n} \mathcal{F}(f \oplus \varphi_\omega) \mathcal{F}(g \oplus \varphi_\omega) \\ 0 &= 2^{2n} + 2^{2n} - 2 \times 2^{2n} \end{aligned} \quad (21)$$

The above equation holds if and only if $\mathcal{F}(f \oplus \varphi_\omega) = \mathcal{F}(g \oplus \varphi_\omega)$ for any $\alpha \in \mathbb{F}_2^n$, if and only if $f(x) = g(x)$.

On the other hand, if $f(x) = g(x)$, then $\triangle_f(\alpha) = \triangle_{f,g}(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$.

(2) By the same method with (1), this result can be proven.

(3) On one hand, according to $\triangle_{f,g}(\alpha) = \triangle_{f,h}(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$, we have

$$\begin{aligned} 0 &= \sum_{\alpha \in \mathbb{F}_2^n} \left[\triangle_{f,g}(\alpha) - \triangle_{f,h}(\alpha)\right]^2 \\ &= \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,g}^2(\alpha) + \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,h}^2(\alpha) \\ &\quad - 2 \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,g}(\alpha) \triangle_{f,h}(\alpha) \\ &= \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,g}(\alpha) \triangle_{f,g}(\alpha) + \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,h}(\alpha) \triangle_{f,h}(\alpha) \\ &\quad - 2 \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,f}(\alpha) \triangle_{g,h}(\alpha) \\ &= \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,f}(\alpha) \triangle_{g,g}(\alpha) + \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,f}(\alpha) \triangle_{h,h}(\alpha) \\ &\quad - 2 \sum_{\alpha \in \mathbb{F}_2^n} \triangle_{f,f}(\alpha) \triangle_{g,h}(\alpha) \\ &= \sum_{\alpha \in \mathbb{F}_2^n} \triangle_f^2(\alpha) + \sum_{\alpha \in \mathbb{F}_2^n} \triangle_f^2(\alpha) - 2 \sum_{\alpha \in \mathbb{F}_2^n} \triangle_f(\alpha) \triangle_{g,h}(\alpha) \\ &= 2\left(\sigma_f - \sum_{\alpha \in \mathbb{F}_2^n} \triangle_f(\alpha) \triangle_{g,h}(\alpha)\right). \end{aligned} \quad (22)$$

On the other hand, by the same method and combining the above equality, we have

$$\begin{aligned} &\sum_{\alpha \in \mathbb{F}_2^n} \left[\triangle_{f,f}(\alpha) - \triangle_{g,h}(\alpha)\right]^2 \\ &\quad = 2\left(\sigma_f - \sum_{\alpha \in \mathbb{F}_2^n} \triangle_f(\alpha) \triangle_{g,h}(\alpha)\right) = 0, \end{aligned} \quad (23)$$

and it implies that $\triangle_f(\alpha) = \triangle_{g,h}(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$. According to (2), we have $g(x) = h(x)$. □

Theorem 12. *Let $f_1(x), f_2(x), f_3(x), f_4(x) \in \mathbb{B}_{n-2}$ satisfy conditions $(\star)$ and $(\star\star)$; then*

$$\begin{aligned} \sigma_{f_1} &= \sigma_{f_2} = \sigma_{f_3} = \sigma_{f_4} = \sigma_{f_1,f_2} = \sigma_{f_1,f_3} = \sigma_{f_1,f_4} \\ &= \sigma_{f_2,f_3} = \sigma_{f_2,f_4} = \sigma_{f_3,f_4}. \end{aligned} \quad (24)$$

Proof. According to condition $(\star)$ and Lemma 5, we have

$$\begin{aligned} \sigma_{f_1} = \sigma_{f_2} &= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1}(\alpha) \triangle_{f_1}(\alpha) \\ &= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1}(\alpha) \triangle_{f_2}(\alpha) = \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}^2 = \sigma_{f_1,f_2}. \end{aligned} \quad (25)$$

Table 1: The autocorrelation value distributions of 3-variable Boolean functions [13].

	ACD	$f(x) \in \mathbb{B}_3$	LS
Class 0	(8,8,8,8,8,8,8,8)	0,255	3
Class 1	(8,8,0,0,0,0,0,0)	192,48,12,252,3,243,207,63	1
Class 2	(8,0,8,0,0,0,0,0)	160,80,10,250,5,245,175,95	1
Class 3	(8,0,0,8,0,0,0,0)	96,144,6,246,9,249,111,159	1
Class 4	(8,0,0,0,8,0,0,0)	136,68,34,238,17,221,187,119	1
Class 5	(8,0,0,0,0,8,0,0)	72,132,18,222,33,237,123,183	1
Class 6	(8,0,0,0,0,0,8,0)	40,20,130,190,65,125,235,215	1
Class 7	(8,0,0,0,0,0,0,8)	24,36,66,126,129,189,219,231	1
Class 8	(8,4,4,4,4,4,4,4)	1,2,4,8,16,32,64,128,254,253,251,247,239,223,191,127	0
Class 9	(8,4,4,4,-4,-4,-4,-4)	224,208,176,112,248,244,242,14,241,13,11,7,143,79,47,31	0
Class 10	(8,4,-4,-4,4,4,-4,-4)	200,196,140,76,236,220,50,206,49,205,35,19,179,115,59,55	0
Class 11	(8,-4,4,-4,4,-4,4,-4)	168,84,162,138,42,234,186,174,81,69,21,213,117,93,171,87	0
Class 12	(8,-4,-4,4,-4,4,4,-4)	104,148,146,134,22,214,182,158,97,73,41,233,121,109,107,151	0
Class 13	(8,-4,-4,4,4,-4,-4,4)	152,100,98,70,38,230,118,110,145,137,25,217,185,157,155,103	0
Class 14	(8,-4,4,-4,-4,4,-4,4)	88,164,82,74,26,218,122,94,161,133,37,229,181,173,91,167	0
Class 15	(8,4,-4,-4,-4,-4,4,4)	56,52,44,28,188,124,194,62,193,61,131,67,227,211,203,199	0
Class 16	(8,0,0,0,0,-8,0,0)	116,139,184,71,226,29,46,209	1
Class 17	(8,0,0,0,-8,0,0,0)	120,180,210,30,225,45,75,135	1
Class 18	(8,0,0,0,0,0,-8,0)	228,216,114,78,177,141,27,39	1
Class 19	(8,0,0,0,0,0,0,-8)	212,178,142,113,77,43,23,232	1
Class 20	(8,0,0,-8,0,0,0,0)	172,92,202,58,197,53,163,83	1
Class 21	(8,0,-8,0,0,0,0,0)	108,156,198,54,201,57,99,147	1
Class 22	(8,-8,0,0,0,0,0,0)	106,154,166,86,169,89,149,101	1
Class 23	(8,8,8,8,-8,-8,-8,-8)	240,15	3
Class 24	(8,8,-8,-8,8,8,-8,-8)	204,51	3
Class 25	(8,8,-8,-8,-8,-8,8,8)	60,195	3
Class 26	(8,-8,8,-8,8,-8,8,-8)	170,85	3
Class 27	(8,-8,8,-8,-8,8,-8,8)	90,165	3
Class 28	(8,-8,-8,8,8,-8,-8,8)	102,153	3
Class 29	(8,-8,-8,8,-8,8,8,-8)	150,105	3

By the same method, we have

$$\begin{aligned}
\sigma_{f_1} &= \sigma_{f_3} = \sigma_{f_1,f_3};\\
\sigma_{f_1} &= \sigma_{f_4} = \sigma_{f_1,f_4};\\
\sigma_{f_2} &= \sigma_{f_3} = \sigma_{f_2,f_3};\\
\sigma_{f_2} &= \sigma_{f_4} = \sigma_{f_2,f_4};\\
\sigma_{f_3} &= \sigma_{f_4} = \sigma_{f_3,f_4}.
\end{aligned} \tag{26}$$

Based on condition $(\star\star)$, we have

$$\begin{aligned}
\sigma_{f_1,f_2} &= \sigma_{f_3,f_4},\\
\sigma_{f_1,f_3} &= \sigma_{f_2,f_4},\\
\sigma_{f_1,f_4} &= \sigma_{f_2,f_3},
\end{aligned} \tag{27}$$

$$\sigma_{f_2,f_3} = \sigma_{f_1,f_4}. \tag{28}$$

Based on (26) and (27), we complete this proof. □

Theorem 13. *Let $f_1(x), f_2(x), f_3(x), f_4(x) \in \mathbb{B}_{n-2}$ satisfy the following conditions:*

(1) $\triangle_{f_1}(\alpha) = \triangle_{f_2}(\alpha) = \triangle_{f_3}(\alpha) = \triangle_{f_4}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$.

(2) $\triangle_{f_1,f_2}(\alpha) = \triangle_{f_3,f_4}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$.

Then $\triangle_{f_1,f_3}(\alpha) = \triangle_{f_2,f_4}(\alpha)$ and $\triangle_{f_1,f_4}(\alpha) = \triangle_{f_2,f_3}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$.

Proof. According to (27), we know that $\sigma_{f_1,f_2} = \sigma_{f_3,f_4}$. Note that we also have

$$\begin{aligned}
0 &= \sum_{\alpha\in\mathbb{F}_2^{n-2}} \left[\triangle_{f_1,f_2}(\alpha) - \triangle_{f_3,f_4}(\alpha)\right]^2\\
&= \sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}^2(\alpha) + \sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_3,f_4}^2(\alpha)\\
&\quad - 2\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha)\\
&= \sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}(\alpha)\triangle_{f_1,f_2}(\alpha) - 2\sum_{\alpha\in\mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}^2(\alpha)
\end{aligned}$$

Table 2: The cross-correlation value distributions of Class 22.

	CCD	$f(x) \in \mathbb{B}_3$	*LS*
Class 22-1	(0,0,8,-8,0,0,0,0)	(106,154);(86,89);(166,169);(149,101)	0
Class 22-2	(0,0,0,0,8,-8,0,0)	(106,166);(86,101);(89,149);(154,169)	0
Class 22-3	(0,0,0,0,0,0,-8,8)	(106,86);(154,89);(166,101);(169,149)	0
Class 22-4	(0,0,0,0,0,0,8,-8)	(106,169);(154,166);(86,149);(89,101)	0
Class 22-5	(0,0,0,0,-8,8,0,0)	(106,89);(154,86);(166,149);(169,101)	0
Class 22-6	(-8,8,0,0,0,0,0,0)	(106,149);(154,101);(166,89);(86,169)	1
Class 22-7	(0,0,-8,8,0,0,0,0)	(106,101);(154,149);(166,86);(169,89)	0

$$
\begin{aligned}
&+ \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_3,f_4}(\alpha)\triangle_{f_3,f_4}(\alpha) \\
&= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_1}(\alpha)\triangle_{f_2,f_2}(\alpha) - 2\sigma_{f_1,f_2} \\
&\quad + \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_3,f_3}(\alpha)\triangle_{f_4,f_4}(\alpha) \\
&= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1}(\alpha)\triangle_{f_2}(\alpha) - 2\sigma_{f_1,f_2} \\
&\quad + \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_3}(\alpha)\triangle_{f_4}(\alpha) = 2\sigma_{f_1} - 2\sigma_{f_1,f_2},
\end{aligned} \tag{29}
$$

and it implies that $\sigma_{f_1} = \sigma_{f_1,f_2}$. Based on this result, we have

$$
\begin{aligned}
&\sum_{\alpha \in \mathbb{F}_2^{n-2}} \left[\triangle_{f_1,f_3}(\alpha) - \triangle_{f_2,f_4}(\alpha)\right]^2 \\
&= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}^2(\alpha) + \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_2,f_4}^2(\alpha) \\
&\quad - 2\sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}(\alpha)\triangle_{f_2,f_4}(\alpha) \\
&= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}(\alpha)\triangle_{f_1,f_3}(\alpha) \\
&\quad + \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_2,f_4}(\alpha)\triangle_{f_2,f_4}(\alpha) \\
&\quad - 2\sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_3}(\alpha)\triangle_{f_2,f_4}(\alpha) \\
&= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_1}(\alpha)\triangle_{f_3,f_3}(\alpha) \\
&\quad + \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_2,f_2}(\alpha)\triangle_{f_4,f_4}(\alpha) \\
&\quad - 2\sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha) \\
&= \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1}(\alpha)\triangle_{f_3}(\alpha) - 2\sigma_{f_1,f_2} \\
&\quad + \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_2}(\alpha)\triangle_{f_4}(\alpha) = 2\sigma_{f_1} - 2\sigma_{f_1,f_2} = 0.
\end{aligned} \tag{30}
$$

It implies that $\triangle_{f_1,f_3}(\alpha) = \triangle_{f_2,f_4}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$.

By the same method, we have $\triangle_{f_1,f_4}(\alpha) = \triangle_{f_2,f_3}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$. □

Based on Theorem 13, we have the following.

Corollary 14. *Let $f_1(x), f_2(x), f_3(x), f_4(x) \in \mathbb{B}_{n-2}$. Conditions $(\star)$ and $(\star\star)$ are equivalent to the following:*

(a) $\triangle_{f_1}(\alpha) = \triangle_{f_2}(\alpha) = \triangle_{f_3}(\alpha) = \triangle_{f_4}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$;

(b) $\triangle_{f_1,f_2}(\alpha) = \triangle_{f_3,f_4}(\alpha)$ for any $\alpha \in \mathbb{F}_2^{n-2}$.

Based on Theorems 9 and 13, we can give an algorithm for finding all (f_1, f_2, f_3, f_4)-pairs satisfying (a) and (b).

Algorithm 15. This algorithm has three steps.

Step 1. We should firstly obtain a set $T_n(f)$, because $T_n(f)$ is a function set with the same autocorrelation distribution. It is implied that we must calculate all autocorrelation distributions with n-variable.

For $n = 3$, [13] gives all autocorrelation distributions (see Table 1); there are 30 different autocorrelation distributions, where *ACD* expresses the autocorrelation distributions.

Step 2. Calculate cross-correlation distributions between any two Boolean functions with the same autocorrelation distribution.

For $n = 3$, there are 30 different distributions of Table 1. In particular, Class 22 has 8 Boolean functions (106, 154, 166, 86, 169, 89, 149, 101) with the same autocorrelation distribution $(8, -8, 0, 0, 0, 0, 0, 0)$. We can calculate all cross-correlation distributions in Table 2. There are 7 different cross-correlation distributions, where *CCD* expresses the cross-correlation distributions.

Step 3. Calculate the number of (f_1, f_2, f_3, f_4)-pairs satisfying (a) and (b) in Theorem 9.

Table 3: The number of (f_1, f_2, f_3, f_4) with 3-variable satisfying conditions $(\star)$ and $(\star\star)$.

	(f_1, f_2, f_3, f_4)	The number of (f_1, f_2, f_3, f_4)-pairs	Class
1	(f_1, f_1, f_1, f_1)	256	Class 0 to 29
2	$(f_1, f_1, f_2, f_2), f_1 \neq f_2$	1360	Class 0 to 29
3	$(f_1, f_2, f_3, f_4) \in A^1$	1316	Class 1 to 22

[1] $A = \{(f_1, f_2, f_3, f_4) : f_1 \neq f_2, f_1 \neq f_3, f_1 \neq f_4, f_2 \neq f_3, f_2 \neq f_4, f_3 \neq f_4\}$.

Table 4: The cross-correlation value distributions of any two 4-variable bent functions.

Cases	*TCCV*	*N*	*Classes*	$\sigma_{f,g}$	$\triangle_{f,g}$
1	[16(1), 0(15)]	6720	15 (448)	256	16
2	[-16(1), 0(15)]	7168	16 (448)	256	16
3	[8(3), -8(1), 0(12)]	107520	560 (192)	256	8
4	[8(1), -8(3), 0(12)]	107520	560 (192)	256	8
5	[4(10), -4(6)]	86016	448 (192)	256	4

(1) *TCCV* is the times of cross-correlation value; for example, in line 1, [16(1), 0(15)] implies that the cross-correlation value 16 occurs one time and the cross-correlation value 0 occurs 15 times; for example, the cross-correlation distributions $(0, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ and $(0, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ all belong to [16(1), 0(15)].
(2) N is the number of pairs $(f(x), g(x))$ with the same *TCCV*.
(3) *Classes* express the number of each class. For example, in line 1, 15(448) denotes that 6720 can be classified into 15 classes (each class has the same cross-correlation distribution) and each class has 448 pairs (the number of any two different bent functions which have the same cross-correlation is 448).

Remark 16. Based on Theorem 13, we analyze the upper bound on σ_f in Theorem 9; this upper bound can be reached if and only if $f_1, f_2, f_3, f_4 \in \mathbb{B}_{n-2}$ satisfying conditions (a) and (b). We must consider the existence of f_1, f_2, f_3, f_4 satisfying conditions (a) and (b).

(1) Suppose that $f_1 = f_2 = f_3 = f_4$. Then conditions (a) and (b) hold.

(2) Suppose that $f_1 = f_2 = f_3 \neq f_4$. According to $\triangle_{f_1,f_2}(\alpha) = \triangle_{f_3,f_4}(\alpha)$, for any $\alpha \in \mathbb{F}_2^n$, we have $\triangle_{f_3}(\alpha) = \triangle_{f_3,f_4}(\alpha)$, for any $\alpha \in \mathbb{F}_2^n$; we know that $f_3 = f_4$. Thus, this supposition cannot be reached.

(3) Suppose that $f_1 = f_2 \neq f_3 = f_4$. Conditions (a) and (b) are equivalent to $\triangle_{f_1}(\alpha) = \triangle_{f_2}(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$; obviously this holds.

(4) Suppose that f_1, f_2, f_3, f_4 are unequal to each other. We can find these (f_1, f_2, f_3, f_4)-pairs satisfying (a) and (b).

For $n = 3$, we give all Boolean functions satisfying conditions (a) and (b) in Example 17.

Example 17. (1) When $n = 3$, for *Class* 22, we will give all Boolean functions satisfying conditions (1), (2), and (4) of Remark 10 in Table 2; it implies that there are 4 pairs of Boolean functions with the same cross-correlation distribution.

(2) In Table 2, the number of Boolean functions satisfying condition (4) is 14. The number of Boolean functions satisfying condition (2) is $\binom{8}{2} = 28$. The number of Boolean functions satisfying condition (1) is $\binom{8}{1} = 8$. Thus, we find 50 (=14+28+8) Boolean functions with 3-variable satisfying (a) and (b); based on the 50 Boolean functions, we can construct 50 5-variable Boolean functions reaching the upper bound in Theorem 9 in Class 22.

(3) The rest of results from Class 0 to Class 29 can be found in Appendix (Tables 6, 7, 8, 9, 10, and 11).

Thus, we find 2932(= 256 + 1360 + 1316) Boolean functions (f_1, f_2, f_3, f_4) satisfying conditions (a) and (b) in all 3-variable Boolean functions in Table 3. By the same method, we also find many Boolean function pairs for $n \geq 4$.

Summary 2. Theorem 13 gives a theoretical result for finding Boolean functions with the same autocorrelation (or cross-correlation) distributions, but Algorithm 15 gives a specific implementation method for lightweight cryptographic decompositions in the Internet of Things [1]. According to Algorithm 15, we can find many Boolean functions with the same *GAC* and the same resistance to attacks.

5. Construction n-Variable Boolean Functions with Lower σ_f by Disjoint Spectrum Functions

Zhang and Zheng [4] showed that the smaller σ_f, the better the *GAC* of a function $f(x) \in \mathbb{B}_n$. In Lemma 6 and Theorem 9, we know that

$$\sigma_f = \sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4} + 6 \sum_{1 \leq i < j \leq 4} \sigma_{f_i, f_j} + 24 \sum_{\alpha \in \mathbb{F}_2^{n-2}} \triangle_{f_1,f_2}(\alpha) \triangle_{f_3,f_4}(\alpha) \quad [14]. \tag{31}$$

Thus, if four decomposition functions f_1, f_2, f_3, f_4 satisfy the conditions,

(1) $\sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4}$ is small;

(2) f_1, f_2, f_3, and f_4 are perfectly uncorrelated,

then f has lower σ_f. It implies that (1) and (2) are important for constructing a good Boolean function f with lower σ_f by the decomposition Boolean functions.

Table 5: Comparison among balanced Boolean functions.

Constructions	n even	N_f	$\triangle_f$	σ_f
[3]	$n \geq 8$	$2^{n-1} - 2^{n/2}$	2^n	2^{2n+2}
[19]	$n \geq 4$	2^{n-2}	2^n	2^{3n-2}
[19]	$n \geq 8$	$2^{n-1} - 2^{n/2}$	2^n	2^{2n+2}
[20]	$n \geq 8$	$2^{n-1} - 2^{n/2}$	–	2^{2n+2}
Theorem 20	$n \geq 6$	$2^{n-1} - 2^{n/2}$	2^n	$5 \cdot 2^{2n-1}$

Table 6: The cross-correlation value distributions of Class 0 and from Class 23 to 29.

	CCD	$f(x) \in \mathbb{B}_3$	*LS*
Class 0	(-8,-8,-8,-8,-8,-8,-8,-8)	(0,255)	3
Class 23	(-8,-8,-8,-8,8,8,8,8)	(240,15)	3
Class 24	(-8,-8,8,8,-8,-8,8,8)	(204,51)	3
Class 25	(-8,-8,8,8,8,8,-8,-8)	(60,195)	3
Class 26	(-8,8,-8,8,-8,8,-8,8)	(170,85)	3
Class 27	(-8,8,-8,8,8,-8,8,-8)	(90,165)	3
Class 28	(-8,8,8,-8,-8,8,8,-8)	(102,153)	3
Class 29	(-8,8,8,-8,8,-8,-8,8)	(150,105)	3

Theorem 9 implies that $\sigma_f = \sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4}$, if any two Boolean functions among f_1, f_2, f_3, and f_4 are perfectly uncorrelated.

Sarkar and Maitra [16] obtained the characterization of perfect uncorrelated.

Lemma 18 (see [16]). *Let $f(x), g(x) \in \mathbb{B}_n$. Then $f(x)$ and $g(x)$ are perfectly uncorrelated if and only if $\mathcal{F}(f \oplus \varphi_\alpha)\mathcal{F}(g \oplus \varphi_\alpha) = 0$ for any $\alpha \in \mathbb{F}_2^n$.*

Disjoint spectra functions have good properties; here we first give a brief summary of pervious results related to the disjoint spectra functions.

(1) Two Boolean functions with disjoint spectra can be used to construct highly nonlinear resilient functions as clearly mentioned in [17].

(2) In 2009, [12] constructed almost optimal resilient functions with even large variables by disjoint spectra functions.

(3) Reference [4] implied $2^{2n} \leq \sigma_f \leq 2^{3n}$ for a Boolean function $f(x) \in \mathbb{B}_n$; $\sigma_f = 2^{2n}$ if and only if $f(x)$ is a Bent function. In order to construct a Boolean function with lower sum-of-squares indicator; therefore [15] constructed a Boolean function $f(x) \in \mathbb{B}_n$ with lower σ_f based on modifying Bent functions and disjoint spectra functions.

Although many authors give constructions with some cryptology properties based on disjoint spectra functions, how to construct disjoint spectra functions which are not (linearly equivalent to) partially linear functions is an open problem [12, 18].

Note that bent functions have minimum sum-of-squares indicator, but any two Bent functions $f(x), g(x) \in \mathbb{B}_n$ are not disjoint spectra functions or perfect uncorrelated. It is because that $|\mathcal{F}(f \oplus \varphi_\alpha)\mathcal{F}(g \oplus \varphi_\alpha)| = 2^n$ for any $\alpha \in \mathbb{F}_2^n$. Thus, $\sigma_{f,g} = 2^{2n}$ and $\#\{\alpha \in \mathbb{F}_2^n : \triangle_{f,g}(\alpha) = 0\} \leq 15$. That is, we cannot construct an n-variable Boolean function by $(n-2)$-variable disjoint spectra bent functions.

In order to construct an n-variable Boolean function $f(x) \in \mathbb{B}_n$ with small σ_f by $(n-2)$-variable bent functions, we give a definition of two pairs of Boolean functions.

Definition 19. Two pairs of n-variable Boolean functions $(f_1(x), f_2(x)), (f_3(x), f_4(x))$ are called to be perfectly uncorrelated if $\triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha) = 0$ for all $\alpha \in \mathbb{F}_2^n$ and are called to be uncorrelated of degree k if $\triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha) = 0$ for all $\alpha \in \mathbb{F}_2^n$ such that $0 \leq wt(\alpha) \leq k$.

Theorem 20. *Let $f(x_n, x_{n-1}, x) = (x_n \oplus 1)(x_{n-1} \oplus 1)f_1(x) \oplus (x_n \oplus 1)x_{n-1}f_2(x) \oplus x_n(x_{n-1} \oplus 1)f_3(x) \oplus x_n x_{n-1} f_4(x)$; $x_n, x_{n-1} \in \mathbb{F}_2$, $x \in \mathbb{F}_2^{n-2}$. If two pairs of $(n-2)$-variable Bent functions $(f_1(x), f_2(x)), (f_3(x), f_4(x))$ are perfectly uncorrelated, then $\sigma_f = 5 \cdot 2^{2n-1}$.*

Proof. According to the above analysis, we know that any two bent functions f_i, f_j $(1 \leq i \neq j \leq 4)$ satisfy $\sigma_{f_i,f_j} = 2^{2(n-2)}$; thus, we have

$$\begin{aligned}\sigma_f &= \sigma_{f_1} + \sigma_{f_2} + \sigma_{f_3} + \sigma_{f_4} + 6\sum_{1\leq i<j\leq 4}\sigma_{f_i,f_j} \\ &\quad + 24\sum_{\alpha\in\mathbb{F}_2^{n-2}}\triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha) \\ &= 40\cdot 2^{2(n-2)} + 24\sum_{\alpha\in\mathbb{F}_2^{n-2}}\triangle_{f_1,f_2}(\alpha)\triangle_{f_3,f_4}(\alpha) \\ &= 40\cdot 2^{2(n-2)} = 5\cdot 2^{2n-1}.\end{aligned} \tag{32}$$

□

Table 7: The cross-correlation value distributions from Class 1 to Class 7.

	CCD	$f(x) \in \mathbb{B}_3$	*LS*
Class 1-1	(0,0,8,8,0,0,0,0)	(192,48);(252,243);(207,63);(12,3)	0
Class 1-2	(0,0,0,0,8,8,0,0)	(192,12);(48,3);(252,207);(243,63)	0
Class 1-3	(0,0,0,0,0,0,-8,-8)	(192,252);(48,243);(12,207);(3,63)	0
Class 1-4	(0,0,0,0,0,0,8,8)	(192,3);(48,12);(252,63);(243,207)	0
Class 1-5	(0,0,0,0,-8,-8,0,0)	(192,243);(48,252);(12,63);(3,207)	0
Class 1-6	(0,0,-8,-8,0,0,0,0)	(192,207);(48,63);(12,252);(3,243)	0
Class 1-7	(-8,-8,0,0,0,0,0,0)	(192,63);(48,207);(12,243);(252,3)	1
Class 2-1	(0,8,0,8,0,0,0,0)	(160,80);(10,5);(250,245);(175,95)	0
Class 2-2	(0,0,0,0,8,0,8,0)	(160,10);(80,5);(250,175);(245,95)	0
Class 2-3	(0,0,0,0,0,-8,0,-8)	(160,250);(80,245);(10,175);(5,95)	0
Class 2-4	(0,0,0,0,0,8,0,8)	(160,5);(80,10);(250,95);(245,175)	0
Class 2-5	(0,0,0,0,-8,0,-8,0)	(160,245);(80,250);(10,95);(5,175)	0
Class 2-6	(0,-8,0,-8,0,0,0,0)	(160,175);(80,95);(10,250);(5,245)	0
Class 2-7	(-8,0,-8,0,0,0,0,0)	(160,95);(80,175);(10,245);(250,5)	1
Class 3-1	(0,8,8,0,0,0,0,0)	(96,144);(6,9);(246,249);(111,159)	0
Class 3-2	(0,0,0,0,8,0,0,8)	(96,6);(144,9);(246,111);(249,159)	0
Class 3-3	(0,0,0,0,0,-8,-8,0)	(96,246);(144,249);(6,111);(9,159)	0
Class 3-4	(0,0,0,0,0,8,8,0)	(96,9);(144,6); (246,159);(249,111)	0
Class 3-5	(0,0,0,0,-8,0,0,-8)	(96,249);(144,246);(6,159);(9,111)	0
Class 3-6	(0,-8,-8,0,0,0,0,0)	(96,111);(144,159);(6,246);(9,249)	0
Class 3-7	(-8,0,0,-8,0,0,0,0)	(96,159);(144,111);(6,249);(246,9)	1
Class 4-1	(0,8,0,0,0,8,0,0)	(136,68);(34,17);(238,221);(187,119)	0
Class 4-2	(0,0,8,0,0,0,8,0)	(136,34);(68,17);(238,187);(221,119)	0
Class 4-3	(0,0,0,-8,0,0,0,-8)	(136,238);(68,221);(34,187);(17,119)	0
Class 4-4	(0,0,0,8,0,0,0,8)	(136,17);(68,34);(238,119);(221,187)	0
Class 4-5	(0,0,-8,0,0,0,-8,0)	(136,221);(68,238);(34,119);(17,187)	0
Class 4-6	(0,-8,0,0,0,-8,0,0)	(136,187);(68,119);(34,238);(17,221)	0
Class 4-7	(-8,0,0,0,-8,0,0,0)	(136,119);(68,187);(34,221);(238,17)	1
Class 5-1	(0,8,0,0,8,0,0,0)	(72,132);(18,33);(222,237);(123,183)	0
Class 5-2	(0,0,8,0,0,0,0,8)	(72,18);(132,33);(222,123);(237,183)	0
Class 5-3	(0,0,0,-8,0,0,-8,0)	(72,222);(132,237);(18,123);(33,183)	0
Class 5-4	(0,0,0,8,0,0,8,0)	(72,33);(132,18);(222,183);(237,123)	0
Class 5-5	(0,0,-8,0,0,0,0,-8)	(72,237);(132,222);(18,183);(33,123)	0
Class 5-6	(0,-8,0,0,-8,0,0,0)	(72,123);(132,183);(18,222);(33,237)	0
Class 5-7	(-8,0,0,0,0,-8,0,0)	(72,183);(132,123);(18,237);(222,33)	1
Class 6-1	(0,8,0,0,0,0,0,8)	(40,20);(130,65);(190,125);(235,215)	0
Class 6-2	(0,0,8,0,8,0,0,0)	(40,130);(20,65);(190,235);(125,215)	0
Class 6-3	(0,0,0,-8,0,-8,0,0)	(40,190);(20,125);(130,235);(65,215)	0
Class 6-4	(0,0,0,8,0,8,0,0)	(40,65);(20,130);(190,215);(125,235)	0
Class 6-5	(0,0,-8,0,-8,0,0,0)	(40,125);(20,190);(130,215);(65,235)	0
Class 6-6	(0,-8,0,0,0,0,0,-8)	(40,235);(130,190);(20,215);(65,125)	0
Class 6-7	(-8,0,0,0,0,0,-8,0)	(40,215);(20,235);(130,125);(190,65)	1
Class 7-1	(0,8,0,0,0,0,8,0)	(24,36);(66,129);(126,189);(219,231)	0
Class 7-2	(0,0,8,0,0,8,0,0)	(24,66);(36,129);(126,219);(189,231)	0
Class 7-3	(0,0,0,-8,-8,0,0,0)	(24,126);(66,219);(36,189);(129,231)	0
Class 7-4	(0,0,0,8,8,0,0,0)	(24,129);(36,66);(126,231);(189,219)	0
Class 7-5	(0,0,-8,0,0,-8,0,0)	(24,189);(66,231);(36,126);(129,219)	0
Class 7-6	(0,-8,0,0,0,0,-8,0)	(24,219);(36,231);(66,126);(129,189)	0
Class 7-7	(-8,0,0,0,0,0,0,-8)	(24,231);(36,219);(66,189);(126,129)	1

Table 8: The cross-correlation value distributions from Class 8 to Class 10.

	CCD	$f(x) \in \mathbb{B}_3$	*LS*
Class 8-1	(4,8,4,4,4,4,4,4)	(1,2);(4,8);(16,32);(64,128);(254,253);(251,247);(239,223);(191,127)	0
Class 8-2	(4,4,8,4,4,4,4,4)	(1,4);(2,8);(16,64);(32,128);(254,251);(253,247);(239,191);(223,127)	0
Class 8-3	(4,4,4,8,4,4,4,4)	(1,8);(2,4);(16,128);(32,64);(254,247);(253,251);(239,127);(223,191)	0
Class 8-4	(4,4,4,4,8,4,4,4)	(1,16);(2,32);(4,64);(8,128);(254,239);(253,223);(251,191);(247,127)	0
Class 8-5	(4,4,4,4,4,8,4,4)	(1,32);(2,16);(4,128);(8,64);(254,223);(253,239);(251,127);(247,191)	0
Class 8-6	(4,4,4,4,4,4,8,4)	(1,64);(2,128);(4,16);(8,32);(254,191);(253,127);(251,239);(247,223)	0
Class 8-7	(4,4,4,4,4,4,4,8)	(1,128);(2,64);(4,32);(8,16);(254,127);(253,191);(251,223);(247,239)	0
Class 8-8	(-8,-4,-4,-4,-4,-4,-4,-4)	(1,254);(2,253);(4,251);(8,247);(16,239);(32,223);(64,191);(128,127)	0
Class 8-9	(-4,-8,-4,-4,-4,-4,-4,-4)	(1,253);(2,254);(4,247);(8,251);(16,223);(32,239);(64,127);(128,191)	0
Class 8-10	(-4,-4,-8,-4,-4,-4,-4,-4)	(1,251);(2,247);(4,254);(8,253);(16,191);(32,127);(64,239);(128,223)	0
Class 8-11	(-4,-4,-4,-8,-4,-4,-4,-4)	(1,247);(2,251);(4,253);(8,254);(16,127);(32,191);(64,223);(128,239)	0
Class 8-12	(-4,-4,-4,-4,-8,-4,-4,-4)	(1,239);(2,223);(4,191);(8,127);(16,254);(32,253);(64,251);(128,247)	0
Class 8-13	(-4,-4,-4,-4,-4,-8,-4,-4)	(1,223);(2,239);(4,127);(8,191);(16,253);(32,254);(64,247);(128,251)	0
Class 8-14	(-4,-4,-4,-4,-4,-4,-8,-4)	(1,191);(2,127);(4,239);(8,223);(16,251);(32,247);(64,254);(128,253)	0
Class 8-15	(-4,-4,-4,-4,-4,-4,-4,-8)	(1,127);(2,191);(4,223);(8,239);(16,247);(32,251);(64,253);(128,254)	0
Class 9-1	(4,8,4,4,-4,-4,-4,-4)	(224,208);(176,112);(248,244);(242,241);(14,13);(11,7);(143,79);(47,31)	0
Class 9-2	(4,4,8,4,-4,-4,-4,-4)	(224,176);(208,112);(248,242);(244,241);(14,11);(13,7);(143,47);(79,31)	0
Class 9-3	(4,4,4,8,-4,-4,-4,-4)	(224,112);(208,176);(248,241);(244,242);(14,7);(13,11);(143,31);(79,47)	0
Class 9-4	(4,4,4,4,-4,-4,-4,-8)	(224,248);(208,244);(176,242);(112,241);(14,143);(13,79);(11,47);(7,31)	0
Class 9-5	(4,4,4,4,-4,-4,-8,-4)	(224,244);(208,248);(176,241);(112,242);(14,79);(13,143);(11,31);(7,47)	0
Class 9-6	(4,4,4,4,-4,-8,-4,-4)	(224,242);(208,241);(176,248);(112,244);(14,47);(13,31);(11,143);(7,79)	0
Class 9-7	(-4,-4,-4,-4,8,4,4,4)	(224,14);(208,13);(176,11);(112,7);(248,143);(244,79);(242,47);(241,31)	0
Class 9-8	(4,4,4,4,-8,-4,-4,-4)	(224,241);(208,242);(176,244);(112,248);(14,31);(13,47);(11,79);(7,143)	0
Class 9-9	(-4,-4,-4,-4,4,8,4,4)	(224,13);(208,14);(176,7);(112,11);(248,79);(244,143);(242,31);(241,47)	0
Class 9-10	(-4,-4,-4,-4,4,4,8,4)	(224,11);(208,7);(176,14);(112,13);(248,47);(244,31);(242,143);(241,79)	0
Class 9-11	(-4,-4,-4,-4,4,4,4,8)	(224,7);(208,11);(176,13);(112,14);(248,31);(244,47);(242,79);(241,143)	0
Class 9-12	(-4,-4,-4,-8,4,4,4,4)	(224,143);(208,79);(176,47);(112,31);(248,14);(244,13);(242,11);(241,7)	0
Class 9-13	(-4,-4,-8,-4,4,4,4,4)	(224,79);(208,143);(176,31);(112,47);(248,13);(244,14);(242,7);(241,11)	0
Class 9-14	(-4,-8,-4,-4,4,4,4,4)	(224,47);(208,31);(176,143);(112,79);(248,11);(244,7);(242,14);(241,13)	0
Class 9-15	(-8,-4,-4,-4,4,4,4,4)	(224,31);(208,47);(176,79);(112,143);(248,7);(244,11);(242,13);(14,241)	0
Class 10-1	(4,8,-4,-4,4,4,-4,-4)	(200,196);(140,76);(236,220);(50,49);(206,205);(35,19);(179,115);(59,55)	0
Class 10-2	(4,4,-4,-4,8,4,-4,-4)	(200,140);(196,76);(236,206);(220,205);(50,35);(49,19);(179,59);(115,55)	0
Class 10-3	(-4,-4,4,8,-4,-4,4,4)	(200,49);(196,50);(140,19);(76,35);(236,115);(220,179);(206,55);(205,59)	0
Class 10-4	(4,4,-8,-4,4,4,-4,-4)	(200,205);(196,206);(140,220);(76,236);(50,55);(49,59);(35,115);(19,179)	0
Class 10-5	(-4,-4,4,4,-4,-4,8,4)	(200,35);(196,19);(140,50);(76,49);(236,59);(220,55);(206,179);(205,115)	0
Class 10-6	(-4,-4,4,4,-4,-4,4,8)	(200,19);(196,35);(140,49);(76,50);(236,55);(220,59);(206,115);(205,179)	0
Class 10-7	(-4,-4,4,4,-4,-8,4,4)	(200,179);(196,115);(140,59);(76,55);(236,50);(220,49);(206,35);(205,19)	0
Class 10-8	(-4,-4,4,4,-8,-4,4,4)	(200,115);(196,179);(140,55);(76,59);(236,49);(220,50);(206,19);(205,35)	0
Class 10-9	(-4,-8,4,4,-4,-4,4,4)	(200,59);(196,55);(140,179);(76,115);(236,35);(220,19);(50,206);(49,205)	0
Class 10-10	(-8,-4,4,4,-4,-4,4,4)	(200,55);(196,59);(140,115);(76,179);(236,19);(220,35);(50,205);(206,49)	0
Class 10-11	(4,4,-4,-4,4,8,-4,-4)	(200,76);(196,140);(236,205);(220,206);(50,19);(49,35);(179,55);(115,59)	0
Class 10-12	(4,4,-4,-4,4,4,-4,-8)	(200,236);(196,220);(140,206);(76,205);(50,179);(49,115);(35,59);(19,55)	0
Class 10-13	(4,4,-4,-4,4,4,-8,-4)	(200,220);(196,236);(140,205);(76,206);(50,115);(49,179);(35,55);(19,59)	0
Class 10-14	(-4,-4,8,4,-4,-4,4,4)	(200,50);(196,49);(140,35);(76,19);(236,179);(220,115);(206,59);(205,55)	0
Class 10-15	(4,4,-4,-8,4,4,-4,-4)	(200,206);(196,205);(140,236);(76,220);(50,59);(49,55);(35,179);(19,115)	0

Table 9: The cross-correlation value distributions from Class 11 to Class 13.

	CCD	$f(x) \in \mathbb{B}_3$	LS
Class 11-1	(-4,8,-4,4,-4,4,-4,4)	(168,84);(162,81);(138,69);(42,21);(234,213);(186,117);(174,93);(171,87)	0
Class 11-2	(4,-4,8,-4,4,-4,4,-4)	(168,162);(84,81);(138,42);(234,186);(174,171);(69,21);(213,117);(93,87)	0
Class 11-3	(4,-4,4,-4,8,-4,4,-4)	(168,138);(84,69);(162,42);(234,174);(186,171);(81,21);(213,93);(117,87)	0
Class 11-4	(4,-4,4,-4,4,-4,8,-4)	(168,42);(84,21);(162,138);(234,171);(186,174);(81,69);(213,87);(117,93)	0
Class 11-5	(4,-4,4,-4,4,-4,4,-8)	(168,234);(84,213);(162,186);(138,174);(42,171);(81,117);(69,93);(21,87)	0
Class 11-6	(4,-4,4,-4,4,-8,4,-4)	(168,186);(84,117);(162,234);(138,171);(42,174);(81,213);(69,87);(21,93)	0
Class 11-7	(4,-4,4,-8,4,-4,4,-4)	(168,174);(84,93);(162,171);(138,234);(42,186);(81,87);(69,213);(21,117)	0
Class 11-8	(-4,4,-4,8,-4,4,-4,4)	(168,81);(84,162);(138,21);(42,69);(234,117);(186,213);(174,87);(93,171)	0
Class 11-9	(-4,4,-4,4,-4,8,-4,4)	(168,69);(84,138);(162,21);(42,81);(234,93);(186,87);(174,213);(117,171)	0
Class 11-10	(-4,4,-4,4,-4,4,-4,8)	(168,21);(84,42);(162,69);(138,81);(234,87);(186,93);(174,117);(213,171)	0
Class 11-11	(-4,4,-4,4,-4,4,-8,4)	(168,213);(84,234);(162,117);(138,93);(42,87);(186,81);(174,69);(21,171)	0
Class 11-12	(-4,4,-4,4,-8,4,-4,4)	(168,117);(84,186);(162,213);(138,87);(42,93);(234,81);(174,21);(69,171)	0
Class 11-13	(-4,4,-8,4,-4,4,-4,4)	(168,93);(84,174);(162,87);(138,213);(42,117);(234,69);(186,21);(81,171)	0
Class 11-14	(4,-8,4,-4,4,-4,4,-4)	(168,171);(84,87);(162,174);(138,186);(42,234);(81,93);(69,117);(21,213)	0
Class 11-15	(-8,4,-4,4,-4,4,-4,5)	(168,87);(84,171);(162,93);(138,117);(42,213);(234,21);(186,69);(174,81)	0
Class 12-1	(-4,8,4,-4,4,-4,-4,4)	(104,148);(146,97);(134,73);(22,41);(214,233);(182,121);(158,109);(107,151)	0
Class 12-2	(-4,4,8,-4,4,-4,-4,4)	(104,146);(148,97);(134,41);(22,73);(214,121);(182,233);(158,107);(109,151)	0
Class 12-3	(-4,4,4,-4,8,-4,-4,4)	(104,134);(148,73);(146,41);(22,97);(214,109);(182,107);(158,233);(121,151)	0
Class 12-4	(-4,4,4,-4,4,-4,-4,8)	(104,22);(148,41);(146,73);(134,97);(214,107);(182,109);(158,121);(233,151)	0
Class 12-5	(-4,4,4,-4,4,-4,-8,4)	(104,214);(148,233);(146,121);(134,109);(22,107);(182,97);(158,73);(41,151)	0
Class 12-6	(-4,4,4,-4,4,-8,-4,4)	(104,182);(148,121);(146,233);(134,107);(22,109);(214,97);(158,41);(73,151)	0
Class 12-7	(-4,4,4,-8,4,-4,-4,4)	(104,158);(148,109);(146,107);(134,233);(22,121);(214,73);(182,41);(97,151)	0
Class 12-8	(4,-4,-4,8,-4,4,4,-4)	(104,97);(148,146);(134,22);(214,182);(158,151);(73,41);(233,121);(109,107)	0
Class 12-9	(4,-4,-4,4,-4,8,4,-4)	(104,73);(148,134);(146,22);(214,158);(182,151);(97,41);(233,109);(121,107)	0
Class 12-10	(4,-4,-4,4,-4,4,8,-4)	(104,41);(148,22);(146,134);(214,151);(182,158);(97,73);(233,107);(121,109)	0
Class 12-11	(4,-4,-4,4,-4,4,4,-8)	(104,233);(148,214);(146,182);(134,158);(22,151);(97,121);(73,109);(41,107)	0
Class 12-12	(4,-4,-4,4,-8,4,4,-4)	(104,121);(148,182);(146,214);(134,151);(22,158);(97,233);(73,107);(41,109)	0
Class 12-13	(4,-4,-8,4,-4,4,4,-4)	(104,109);(148,158);(146,151);(134,214);(22,182);(97,107);(73,233);(41,121)	0
Class 12-14	(4,-8,-4,4,-4,4,4,-4)	(104,107);(148,151);(146,158);(134,182);(22,214);(97,109);(73,121);(41,233)	0
Class 12-15	(-8,4,4,-4,4,-4,-4,4)	(104,151);(148,107);(146,109);(134,121);(22,233);(214,41);(182,73);(158,97)	0
Class 13-1	(-4,8,4,-4,-4,4,4,-4)	(152,100);(98,145);(70,137);(38,25);(230,217);(118,185);(110,157);(155,103)	0
Class 13-2	(-4,4,8,-4,-4,4,4,-4)	(152,98);(100,145);(70,25);(38,137);(230,185);(118,217);(110,155);(157,103)	0
Class 13-3	(-4,4,4,-4,-4,8,4,-4)	(152,70);(100,137);(98,25);(38,145);(230,157);(118,155);(110,217);(185,103)	0
Class 13-4	(-4,4,4,-4,-4,4,4,-8)	(152,230);(100,217);(98,185);(70,157);(38,155);(118,145);(110,137);(25,103)	0
Class 13-5	(-4,4,4,-4,-8,4,4,-4)	(152,118);(100,185);(98,217);(70,155);(38,157);(230,145);(110,25);(137,103)	0
Class 13-6	(-4,4,4,-4,-4,4,8,-4)	(152,38);(100,25);(98,137);(70,145);(230,155);(118,157);(110,185);(217,103)	0
Class 13-7	(-4,4,4,-8,-4,4,4,-4)	(152,110);(100,157);(98,155);(70,217);(38,185);(230,137);(118,25);(145,103)	0
Class 13-8	(4,-4,-4,8,4,-4,-4,4)	(152,145);(100,98);(70,38);(230,118);(110,103);(137,25);(217,185);(157,155)	0
Class 13-9	(4,-4,-4,4,8,-4,-4,4)	(152,137);(100,70);(98,38);(230,110);(118,103);(145,25);(217,157);(185,155)	0
Class 13-10	(4,-4,-4,4,4,-4,-4,8)	(152,25);(100,38);(98,70);(230,103);(118,110);(145,137);(217,155);(185,157)	0
Class 13-11	(4,-4,-4,4,4,-4,-8,4)	(152,217);(100,230);(98,118);(70,110);(38,103);(145,185);(137,157);(25,155)	0
Class 13-12	(4,-4,-4,4,4,-8,-4,4)	(152,185);(100,118);(98,230);(70,103);(38,110);(145,217);(137,155);(25,157)	0
Class 13-13	(4,-4,-8,4,4,-4,-4,4)	(152,157);(100,110);(98,103);(70,230);(38,118);(145,155);(137,217);(25,185)	0
Class 13-14	(4,-8,-4,4,4,-4,-4,4)	(152,155);(100,103);(98,110);(70,118);(38,230);(145,157);(137,185);(25,217)	0
Class 13-15	(-8,4,4,-4,-4,4,4,-4)	(152,103);(100,155);(98,157);(70,185);(38,217);(230,25);(118,137);(110,145)	0

Table 10: The cross-correlation value distributions from Class 14 to Class 15.

	CCD	$f(x) \in \mathbb{B}_3$	*LS*
Class 14-1	(-4,8,-4,4,4,-4,4,-4)	(88,164);(82,161);(74,133);(26,37);(218,229);(122,181);(94,173);(91,167)	0
Class 14-2	(4,-4,8,-4,-4,4,-4,4)	(88,82);(164,161);(74,26);(218,122);(94,91);(133,37);(229,181);(173,167)	0
Class 14-3	(4,-4,4,-4,-4,8,-4,4)	(88,74);(164,133);(82,26);(218,94);(122,91);(161,37);(229,173);(181,167)	0
Class 14-4	(4,-4,4,-4,-4,4,-4,8)	(88,26);(164,37);(82,74);(218,91);(122,94);(161,133);(229,167);(181,173)	0
Class 14-5	(4,-4,4,-4,-4,4,-8,4)	(88,218);(164,229);(82,122);(74,94);(26,91);(161,181);(133,173);(37,167)	0
Class 14-6	(4,-4,4,-4,-8,4,-4,4)	(88,122);(164,181);(82,218);(74,91);(26,94);(161,229);(133,167);(37,173)	0
Class 14-7	(4,-4,4,-8,-4,4,-4,4)	(88,94);(164,173);(82,91);(74,218);(26,122);(161,167);(133,229);(37,181)	0
Class 14-8	(-4,4,-4,8,4,-4,4,-4)	(88,161);(164,82);(74,37);(26,133);(218,181);(122,229);(94,167);(173,91)	0
Class 14-9	(-4,4,-4,4,8,-4,4,-4)	(88,133);(164,74);(82,37);(26,161);(218,173);(122,167);(94,229);(181,91)	0
Class 14-10	(-4,4,-4,4,4,-4,8,-4)	(88,37);(164,26);(82,133);(74,161);(218,167);(122,173);(94,181);(229,91)	0
Class 14-11	(-4,4,-4,4,4,-4,4,-8)	(88,229);(164,218);(82,181);(74,173);(26,167);(122,161);(94,133);(37,91)	0
Class 14-12	(-4,4,-4,4,4,-8,4,-4)	(88,181);(164,122);(82,229);(74,167);(26,173);(218,161);(94,37);(133,91)	0
Class 14-13	(-4,4,-8,4,4,-4,4,-4)	(88,173);(164,94);(82,167);(74,229);(26,181);(218,133);(122,37);(161,91)	0
Class 14-14	(4,-8,4,-4,-4,4,-4,4)	(88,91);(164,167);(82,94);(74,122);(26,218);(161,173);(133,181);(37,229)	0
Class 14-15	(-8,4,-4,4,4,-4,4,-4)	(88,167);(164,91);(82,173);(74,181);(26,229);(218,37);(122,133);(94,161)	0
Class 15-1	(4,8,-4,-4,-4,-4,4,4)	(56,52);(44,28),(188,124);(194,193);(62,61);(131,67);(203,199);(227,211)	0
Class 15-2	(4,4,-4,-4,-4,-4,8,4)	(56,44);(52,28);(188,62);(124,61);(194,131);(193,67);(227,203);(211,199)	0
Class 15-3	(4,4,-4,-4,-4,-4,4,8)	(56,28);(52,44);(188,61);(124,62);(194,67);(193,131);(227,199);(211,203)	0
Class 15-4	(4,4,-4,-4,-4,-8,4,4)	(56,188);(52,124);(44,62);(28,61);(194,227);(193,211);(131,203);(67,199)	0
Class 15-5	(4,4,-4,-4,-8,-4,4,4)	(56,124);(52,188);(44,61);(28,62);(194,211);(193,227);(131,199);(67,203)	0
Class 15-6	(-4,-4,8,4,4,4,-4,-4)	(56,194);(52,193);(44,131);(28,67);(188,227);(124,211);(62,203);(61,199)	0
Class 15-7	(4,4,-4,-8,-4,-4,4,4)	(56,62);(52,61);(44,188);(28,124);(194,203);(193,199);(131,227);(67,211)	0
Class 15-8	(-4,-4,4,8,4,4,-4,-4)	(56,193);(52,194);(44,67);(28,131);(188,211);(124,227);(62,199);(61,203)	0
Class 15-9	(4,4,-8,-4,-4,-4,4,4)	(56,61);(52,62);(44,124);(28,288);(194,199);(193,203);(131,211);(67,227)	0
Class 15-10	(-4,-4,4,4,8,4,-4,-4)	(56,131);(52,67);(44,194);(28,193);(188,203);(124,199);(62,227);(61,211)	0
Class 15-11	(-4,-4,4,4,4,8,-4,-4)	(56,67);(52,131);(44,193);(28,194);(188,199);(124,203);(62,211);(61,227)	0
Class 15-12	(-4,-4,4,4,4,4,-4,-8)	(56,227);(52,211);(44,203);(28,199);(188,194);(124,193);(62,131);(61,67)	0
Class 15-13	(-4,-4,4,4,4,4,-8,-4)	(56,211);(52,227);(44,199);(28,203);(188,193);(124,194);(62,67);(61,131)	0
Class 15-14	(-4,-8,4,4,4,4,-4,-4)	(56,203);(52,199);(44,227);(28,211);(188,131);(124,67);(194,62);(193,61)	0
Class 15-15	(-8,-4,4,4,4,4,-4,-4)	(56,199);(52,203);(44,211);(28,227);(188,67);(124,131);(194,61);(62,193)	0

In Theorem 20, we can give many n-variable Boolean functions by using $(n-2)$-variable decomposition Bent functions.

Example 21. For $n = 4$, we give the cross-correlation value distribution between any two bent functions in Table 4.

(1) We find that the number of the cross-correlation distributions of any two bent functions from 896 bent functions is 2047 (=15+16+560+560+448) classes; that is, there are 2047 different cross-correlation distributions of any two bent functions.

(2) We obtain that the number of perfect uncorrelated pairs is 201210 among 2047 different cross-correlation distributions. It implies that one can find many Bent functions-pairs satisfying disjoint spectrum; that is, we can construct lots of Boolean functions $f(x) \in \mathbb{B}_n$ with $\sigma_f = 5 \cdot 2^{2n-1}$ in Theorem 20. Meanwhile, by Theorem 20, we can obtain many balanced Boolean functions $f(x)$, if bent Boolean functions f_1, f_2, f_3, f_4 satisfy $wt(f_1) + wt(f_2) + wt(f_3) + wt(f_4) = 2^{n-1}$; it is easy to find f_1, f_2, f_3, f_4 satisfying $wt(f_1) + wt(f_2) + wt(f_3) + wt(f_4) = 2^{n-1}$. For example, $wt(f_1) = wt(f_3) = 2^{n-3} + 2^{(n-2)/2-1}$ and $wt(f_2) = wt(f_4) = 2^{n-3} - 2^{(n-2)/2-1}$. Thus, if bent Boolean functions f_1, f_2, f_3, f_4 satisfy the following conditions,

(1) $wt(f_1) + wt(f_2) + wt(f_3) + wt(f_4) = 2^{n-1}$;

(2) two pairs of $(n-2)$-variable Bent functions $(f_1(x), f_2(x)), (f_3(x), f_4(x))$ are perfectly uncorrelated,

then f is a balanced Boolean function with $\sigma_f = 5 \cdot 2^{2n-1}$.

In stream cipher, constructing Boolean function f with high nonlinearity $\mathcal{N}_f$ and very good *GAC* property (low absolute indicator $\triangle_f$ and low sum-of-squares indicator σ_f) is favored. Addressing this problem, many works have been done; see, for instance, [3, 19, 20], which are summarized in Table 5; we find that our result is better than their methods.

Summary 3. Theorem 20 provides a construction for use in lightweight dynamic cryptographic algorithms, especially some 3-variable or 4-variable Boolean functions for encryption algorithm in the Internet of Things [2]; that is, we find many alternative cryptographic components with the same cryptographic properties.

Table 11: The cross-correlation value distributions from Class 16 to Class 21.

	CCD	$f(x) \in \mathbb{B}_3$	LS
Class 16-1	(-8,0,0,0,0,8,0,0)	(116,139);(184,71);(226,29);(46,209)	1
Class 16-2	(0,8,0,0,-8,0,0,0)	(116,184);(139,71);(226,209);(29,46)	0
Class 16-3	(0,-8,0,0,8,0,0,0)	(116,71);(226,46);(29,209);(139,184)	0
Class 16-4	(0,0,0,8,0,0,-8,0)	(116,226);(139,29);(184,209);(71,46)	0
Class 16-5	(0,0,0,-8,0,0,8,0)	(116,29);(139,226);(184,46);(71,209)	0
Class 16-6	(0,0,-8,0,0,0,0,8)	(116,46);(139,209);(184,29);(71,226)	0
Class 16-7	(0,0,8,0,0,0,0,-8)	(116,209);(139,46);(184,226);(71,29)	0
Class 17-1	(0,8,0,0,0,-8,0,0)	(120,180);(210,225);(30,45);(75,135)	0
Class 17-2	(0,0,8,0,0,0,-8,0)	(120,210);(180,225);(30,75);(45,135)	0
Class 17-3	(0,0,0,-8,0,0,0,8)	(120,30);(180,45);(210,75);(225,135)	0
Class 17-4	(0,0,0,8,0,0,0,-8)	(120,225);(180,210);(30,135);(45,75)	0
Class 17-5	(0,0,-8,0,0,0,8,0)	(120,45);(180,30);(210,135);(225,75)	0
Class 17-6	(0,-8,0,0,0,8,0,0)	(120,75);(180,135);(210,30);(225,45)	0
Class 17-7	(-8,0,0,0,8,0,0,0)	(120,135);(180,75);(210,45);(30,225)	1
Class 18-1	(0,8,0,0,0,0,0,-8)	(228,216);(114,177);(78,141);(27,39)	0
Class 18-2	(0,0,0,8,0,-8,0,0)	(228,114);(78,39);(141,27);(216,177)	0
Class 18-3	(0,0,-8,0,8,0,0,0)	(228,78);(114,39);(177,27);(216,141)	0
Class 18-4	(0,0,8,0,-8,0,0,0)	(228,177);(216,114);(78,27);(141,39)	0
Class 18-5	(0,0,0,-8,0,8,0,0)	(228,141);(216,78);(114,27);(177,39)	0
Class 18-6	(-8,0,0,0,0,0,8,0)	(228,27);(216,39);(114,141);(78,177)	1
Class 18-7	(0,-8,0,0,0,0,0,8)	(228,39);(216,27);(114,78);(177,141)	0
Class 19-1	(0,0,0,8,-8,0,0,0)	(212,178);(142,23);(113,232);(77,43)	0
Class 19-2	(0,0,-8,0,0,8,0,0)	(212,142);(178,23);(113,43);(77,232)	0
Class 19-3	(0,0,8,0,0,-8,0,0)	(212,113);(178,232);(142,43);(77,23)	0
Class 19-4	(0,0,0,-8,8,0,0,0)	(212,77);(178,43);(142,232);(113,23)	0
Class 19-5	(-8,0,0,0,0,0,0,8)	(212,43);(178,77);(142,113);(23,232)	1
Class 19-6	(0,-8,0,0,0,0,8,0)	(212,23);(178,142);(113,77);(43,232)	0
Class 19-7	(0,8,0,0,0,0,-8,0)	(212,232);(178,113);(142,77);(43,23)	0
Class 20-1	(0,8,-8,0,0,0,0,0)	(172,92);(163,83);(202,197);(58,53)	0
Class 20-2	(0,0,0,0,8,0,0,-8)	(172,202);(92,197);(58,163);(53,83)	0
Class 20-3	(0,0,0,0,0,-8,8,0)	(172,58);(92,53);(202,163);(197,83)	0
Class 20-4	(0,0,0,0,0,8,-8,0)	(172,197);(92,202);(58,83);(53,163)	0
Class 20-5	(0,0,0,0,-8,0,0,8)	(172,53);(92,58);(202,83);(197,163)	0
Class 20-6	(0,-8,8,0,0,0,0,0)	(172,163);(92,83);(202,58);(197,53)	0
Class 20-7	(-8,0,0,8,0,0,0,0)	(172,83);(92,163);(202,53);(58,197)	1
Class 21-1	(0,8,0,-8,0,0,0,0)	(108,156);(198,201);(54,57);(99,147)	0
Class 21-2	(0,0,0,0,8,0,-8,0)	(108,198);(156,201) (54,99);(57,147)	0
Class 21-3	(0,0,0,0,0,-8,0,8)	(108,54);(156,57);(198,99);(201,147)	0
Class 21-4	(0,0,0,0,0,8,0,-8)	(108,201);(156,198);(54,147);(57,99)	0
Class 21-5	(0,0,0,0,-8,0,8,0)	(108,57);(156,54);(198,147);(201,99)	0
Class 21-6	(0,-8,0,8,0,0,0,0)	(108,99);(156,147);(198,54);(201,57)	0
Class 21-7	(-8,0,8,0,0,0,0,0)	(108,147);(156,99);(198,57);(54,201)	1

6. Conclusions

In this paper, we have derived a construction method to obtain a Boolean function with small sum-of-squares indicator by decomposition Boolean functions; some properties and a search algorithm of Boolean functions with the same autocorrelation (or cross-correlation) distribution are given. We put up a new definition of two pairs of Boolean functions; this definition plays an important role in our construction. We believe that these conclusions and properties can be widely studied in designing the stream ciphers and block ciphers. In particular, Boolean functions with the same autocorrelation (or cross-correlation) distribution provide optional components for lightweight cryptographic

algorithms in the Internet of Things. Using these Boolean functions makes cryptographic algorithms dynamic but does not change the security strength of cryptographic algorithms.

Appendix

See Tables 6, 7, 8, 9, 10, and 11.

Data Availability

All data used to support the findings of this study are included within the article (e.g., Table 1, Table 2, ⋯, Table 11); no other data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest in the publication of this paper.

Acknowledgments

The first author is supported by the National Key R&D Program of China (nos. 2017YFB0802000 and 2017YFB0802004). The second author is supported by the National Key R&D Program of China (no. 2017YFB0802000) and in part by the National Natural Science Foundation of China (nos. 61572148 and 61872103). The last author is supported by Jiangsu Natural Science Foundation (BK20181352) and in part by the Fundamental Research Funds for the Central Universities (no. 2015XKMS058).

References

[1] D. Evans, *The Internet of Things How the next Evolution of the Internet is Changing Everything*, Cisco Internet Business Solutions Group (IBSG), April 2011, https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[2] K. Brockmeier, "Gartner Adds Big Date, Gamification, and Internet of Things to Its Hype Cycle, Read Write Enterprise," *Trend Analysis*, 2011, https://readwrite.com/2011/08/11/gartner-adds-big-data-gamifica/.

[3] A. Canteaut, C. Carlet, P. Charpin, and C. Fontaine, "Propagation characteristics and correlation immunity of highly nonlinearity Boolean functions," in *Advances in Cryptology - EUROCRYPT 2000*, vol. 1807 of *Lecture Notes in Computer Sceince*, pp. 507–522, Springer, Berlin, 2000.

[4] X.-M. Zhang and Y. Zheng, "GAC- the criterion for global avalanche characteristics of cryptographic functions," *The Journal of Universal Computer Science*, vol. 1, no. 5, pp. 320–337, 1995.

[5] N. T. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback," in *Advances in cryptology—EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Sceince*, pp. 345–359, Springer, Berlin, 2003.

[6] Q. Wang and P. Stănică, "Transparency order for Boolean functions: analysis and construction," *Designs, Codes and Cryptography*, pp. 1–17, 2019.

[7] J. J. Son, J. I. Lim, S. Chee, and S. H. Sung, "Global avalanche characteristics and nonlinearity of balanced Boolean function," *Information Processing Letters*, vol. 65, no. 3, pp. 139–144, 1998.

[8] S. H. Sung, S. Chee, and C. Park, "Global avalanche characteristics and propagation criterion of balanced Boolean functions," *Information Processing Letters*, vol. 69, no. 1, pp. 21–24, 1999.

[9] Y. Zhou, M. Xie, and G. Xiao, "On the global avalanche characteristics of two Boolean functions and the higher order nonlinearity," *Information Sciences*, vol. 180, no. 2, pp. 256–265, 2010.

[10] Y. Zhou, X. Dong, W. Zhang, and B. Zeng, "New bounds on the sum-of-squares indicator," in *Proceedings of the 7th International ICST Conference on Communications and Networking in China (CHINACOM '12)*, pp. 173–178, Kun Ming, August 2012.

[11] D. Tang, W. Zhang, and X. Tang, "Construction of balanced Boolean functions with high nonlinearity and good auto-correlation properties," *Designs, Codes and Cryptography. An International Journal*, vol. 67, no. 1, pp. 77–91, 2013.

[12] W. Zhang and G. Xiao, "Constructions of almost optimal resilient Boolean functions on large even number of variables," *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5822–5831, 2009.

[13] Y. Zhou, "On the distribution of auto-correlation value of balanced Boolean functions," *Advances in Mathematics of Communications*, vol. 7, no. 3, pp. 335–347, 2013.

[14] Z. Zhuo, J. Chong, R. Yu, and M. Ren, "Global avalanche characteristics of Boolean functions by concatenation," *Journal of Harbin Institute of Technology (New Series)*, vol. 23, no. 3, pp. 91–96, 2016.

[15] Y. Zhou, W. Zhang, S. Zhu, and G. Xiao, "The global avalanche characteristics of two Boolean functions and algebraic immunity," *International Journal of Computer Mathematics*, vol. 89, no. 16, pp. 2165–2179, 2012.

[16] P. Sarkar and S. Maitra, "Cross-correlation analysis of cryptographically useful Boolean functions and S-boxes," *Theory of Computing Systems*, vol. 35, no. 1, pp. 39–57, 2002.

[17] E. Pasalic, S. Maitra, T. Johansson, and P. Sarkar, "New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity," in *Proceedings of the Workshop o Coding and Cryptography - WCC '01*, vol. 6 of *Electronic Notes in Discrete Mathematics*, pp. 158–167, Amsterdam, The Netherlands: Elsevier Science, Paris, France, 2001.

[18] F. Zhang, Y. Wei, E. Pasalic, and S. Xia, "Large sets of disjoint spectra plateaued functions inequivalent to partially linear functions," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 64, no. 4, part 2, pp. 2987–2999, 2018.

[19] P. Stănică, "Nonlinearity, local and global avalanche characteristics of balanced Boolean functions," *Discrete Mathematics*, vol. 248, no. 1-3, pp. 181–193, 2002.

[20] P. Stănică and S. H. Sung, "Improving the nonlinearity of certain balanced Boolean functions with good local and global avalanche characteristics," *Information Processing Letters*, vol. 79, no. 4, pp. 167–172, 2001.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 9078176, 15 pages
https://doi.org/10.1155/2019/9078176

Research Article

MSFA: Multiple System Fingerprint Attack Scheme for IoT Anonymous Communication

Tianbo Lu,[1,2] **Ting Meng**,[1,2] **Chao Li**,[3] **Guozhen Dong**,[1,2] **Huiyang Li**,[4] **Jiao Zhang**,[1] **and Xiaoyan Zhang**[1]

[1] *School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Key Laboratory of Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunication, Ministry of Education, Beijing 100876, China*
[3] *Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China*
[4] *Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, VA 76091, USA*

Correspondence should be addressed to Tianbo Lu; lutb@bupt.edu.cn and Chao Li; lichao@gzhu.edu.cn

Received 22 January 2019; Accepted 27 February 2019; Published 27 March 2019

Guest Editor: Fagen Li

For the past few years, Internet of Things (IoT) has developed rapidly and been extensively used. However, its transmission security and privacy protection are insufficient, which limits the development of IoT to a certain extent. As a technology of IoT information transmission, anonymous communication technology comes into being as an important means to ensure the security of healthcare data, which can better protect users' privacy in some ways. Nowadays, a variety of attack techniques for anonymous communication systems have been proposed by the academic community to track senders and receivers or discover communications between two users. Thus, the MSFA (Multiple System Fingerprint Attack) scheme for anonymous communication systems is presented in this paper where the MSFA scheme architecture, implementation in the Tor environment, and experimental data processing are described. Through a comparative analysis between two traces of visiting the same website based on the edit distance, it is shown that the longer the length of the site traffic data, the greater the edit distance of the site access traffic and the larger the range.

1. Introduction

A great number of research achievements have been gained about anonymous communication systems in the past 40 years. Ruei-Hau Hsu et al. [1] proposed network-covered and network-absent authenticated key exchange protocols for D2D communications to guarantee accountable group anonymity, end-to-end security to network operators. Anonymous communication technology is an effective technique for healthcare data privacy protection. Amin et al. [2] proposed the architecture of a patient monitoring system in WMSN (wireless medical sensor network) and designed an anonymous mutual authentication protocol suitable for mobile users to provide secure access and privacy for patient data. Mingshan Xie et al. [3] proposed the anonymization protection algorithm which is suitable for the data exchange in an incompletely open manner for the ego of data in the IoT. The anonymous dataset generated by the algorithm can effectively protect the sensitive information of IoT under the premise of ensuring the availability of the data. The EXCHANge protocol [4], a cryptoless over-the-air key establishment multiround protocol based on sender/receiver anonymity, was specifically conceived to secure IoT networks based on the IEEE 802.15.4 communication technology. Network malicious attackers or criminals rarely attack directly through their own computers. Before attacking the final target, they often land anonymous communication systems to hide their identities, such as Tor [5], JAP [6], Freenet [7], and I2P [8]. Tor and I2P have a large group of users and they have published software versions for mobile ad-hoc networks. The correspondence between input and output streams is hidden by Tor anonymous system in a variety of ways, while the attacker's goal is to identify the correspondence. Almuhtadi et al. [9] made a preliminary evaluation about Misty

clouds which is a privacy-preserving platform for online user anonymity in Social Internet of Things, indicating that the new algorithm was better than the existing Tor algorithm and could achieve the expected privacy goal within the expected performance cost. The communication relationship between sender and receiver in the anonymous communication network can be discovered by the network fingerprint attack. The academia has carried out extensive research on the scheme and application of fingerprint attack.

Network traffic can be disturbed; for example, packets may be cached on a relay node for a period of time or be cut, recombined, retransmitted, or even lost. In web fingerprint, passive traffic analysis attack techniques are used that only require an attacker to configure a network environment similar to a regulator and access the target site using the same encryption proxy technology. The actual address of the regulator's communication side is identified by analyzing the generated traffic characteristics. Fingerprinting [10] combines a number of input sources. Fingerprint attack technique identifies whether the sender communicates with a particular recipient by collecting the sender or receiving the feature information of both. The feature information can be network traffic characteristics, routing information characteristics, and node information characteristics. When the receiver communicates with the sender, the feature information between them will be collected by the fingerprint technology to form the fingerprint, which can determine a communication relationship between the sender and the receiver when they communicate again. In the existing fingerprint identification attacks [11–18], the researchers use the packet size distribution, the sum of the packet size, packet timing interval, etc. as the basic statistical features to characterize the web fingerprint feature set. Researchers have demonstrated the feasibility of the website fingerprint attack methods and conducted further research on web fingerprinting. Cai et al. [11] proposed a website fingerprint attack that could successfully attack the latest proposed defensive traffic analysis attack scheme HTTPOS [19].

Our contributions are as follows: based on the CAI fingerprint attack prototype [11], we propose the MSFA fingerprint attack scheme in three aspects, namely, MSFA scheme architecture and module design, the implementation of MSFA scheme in Tor anonymous communication network, and the capture of the original traffic information and data processing.

2. CAI Fingerprint Attack Scheme

In this section, the background of the CAI fingerprint attack scenario, the CAI fingerprint attack model, as well as the attack process and features will be outlined.

2.1. The Background of the CAI Fingerprint Attack Scheme. Cai et al. [11] proposed a website fingerprint attack that could successfully attack the HTTPOS [13] traffic analysis scheme. CAI fingerprint attack is based on a simple network behavior model which can correctly predict the pages accessed by users over half of the time for any defense model. At the same time, it can correctly identify whether the user accesses a specific site with the experimental success rate over 90%.

2.1.1. Web Page Tracking. Web pages contain multiple objects such as HTML files, images, and flash, and the browser sends a separate request to each object. With the way of a combination of multiple TCP links and pipelines, it is more quickly for browsers to load pages. The browser requests the page-related objects before loading the page. Note that the order of requests has inheritance stability, and an object can only be requested after the browser has received some referencing pages. Some requests may be delayed due to the CPU load and packet reordering so that the order of requests and responses may be different when the browser loads a page every time. Some requests may be omitted if there is a copy of the object in memory. The number of requests sent by the browser and the total number of packets returned to the server may vary with the change of the size of the dynamic web page and the objects it contains [11].

2.1.2. Damerau-Levenshtein Edit Distance. In information theory and computer science, Damerau-Levenshtein distance [20–22] indicates the distance between two strings. In short, it refers to two finite sequences of symbols that convert a string to another string with a minimum number of operations, where the operation is defined as an insertion, deletion, replacement, or swap of two adjacent characters. In Damerau's [20] study, not only are the four edits distinguished, but it also points out that they correspond to more than 80% of all spelling errors while Damerau only considers edits that could correct a misspelling. The difference between Damerau-Levenshtein and Classic Levenshtein is that Damerau-Levenshtein distance allows the exchange between characters while only insert, delete, and replace operations are permitted in Levenshtein distances. The Levenshtein distance is optimized to include the exchange of adjacent characters, resulting in the different measurement distances called the Damerau-Levenshtein distance [21].

2.1.3. LIBSVM. LIBSVM software is a library for integrated support vector machines that supports multicategory classification. Support vector machine (SVM) is a technology that effectively classifies data. The essence of LIBSVM is a library for support vector machines, and there is no need for users to understand the basic theory behind the support vector machines while just following the basic program to get the corresponding result. A classification task for LIBSVM usually involves two separate datasets, a training set and a testing set. A model can be generated by LIBSVM based on the training data, which predicts the target value of the test set. The data in the test set only provides the attribute values of the test data.

2.2. The Characteristics of CAI Fingerprint Attack. In the CAI fingerprint attack process (Figure 1), the client traffic

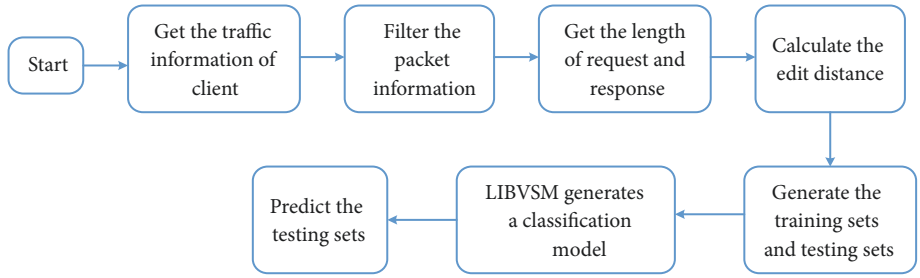


Figure 1: CAI fingerprint attack process.

information is captured through the client agent first and then the captured packet information is processed to obtain the packet length information. The edit distance is then calculated to generate the training and testing sets required for LIBSVM classification. The classification model is generated by LIBSVM for the prediction of the test set data to determine whether the user has visited the target website.

Firstly, Cai et al. proposed a new method to calculate the similarity of web access [11]. The order of the request and response packets shows the size and importance of the objects referenced in a page, so packet scheduling is very important for identifying web pages. CAI fingerprint attack transformed tracks into strings and compared the similarities between the two tracks using edit distances. Damerau-Levenshtein distance is a good metric that allows insertion, deletion, replacement, and interchange. Secondly, the CAI fingerprint attack scheme used LIBSVM to establish a classification model for the processed web packet data and predict the pages accessed by the user.

3. MSFA Attack Scheme

In this section, we have designed the MSFA fingerprint attack scheme.

3.1. MSFA Scheme Architecture. The architecture of the MSFA fingerprint attack scheme which includes the design goals of the MSFA scheme, the threat model, and the attack model is discussed.

3.1.1. MSFA Design Goal. The design idea of the MSFA attack scheme is based on CAI fingerprint attack prototype, and the concrete realization scheme is proposed for different anonymous systems. The following describes the MSFA attack scheme through three aspects:

(i) The database system of the MASF attack scheme. The captured network traffic information can be effectively saved by the database system, which designs a scientific and reasonable database for the site IP address, traffic information, site name, etc. The database system is provided with the advantages of simple structure and clear function.

(ii) The improved MSFA attack scheme is based on the CAI fingerprint attack scheme, and experiments are performed on different categories of websites for data collection. The MSFA attack scheme is implemented in the Tor anonymous communication system environment.

(iii) Capture the original flow of information. To obtain the fingerprint information of the website, it is necessary to capture the website original traffic information. A web page is composed of multiple objects, and a separate request is sent for each object by the browser. Multiple TCP links can be exploited by the browser to load the page faster. The browser will request the relevant objects of the page before loading the page that generates network traffic. The packets in the tracking can be roughly divided into two categories: request packets and response packets. When a user browses the encrypted proxy page, all the relevant documents on the page will be downloaded by the user browser and each of them requires a separate TCP link to return.

3.1.2. MSFA Scheme Threat Model. As shown in Figure 2, firstly, it needs to connect the I2P network through the local connection when Amy visits the website through the I2P network. Kad algorithm is exploited by the I2P network to obtain information on the network node and access the destination server through the nodes of the I2P network. Multiple links are used by the I2P to send and receive data, but if the links of sending data and receiving data are different, the number of nodes on the two links will be different. Ken accesses through the Tor network and it reaches the destination server to visit the site through the entrance node, intermediate node, and exit node three hops in the Tor network. The three-hop routing nodes which have been set up will not automatically change unless they are manually changed. Moreover, the I2P routing nodes have a valid period. Fingerprint attacker captures the traffic between the client and the anonymous communication system entry node, subsequently analyzes the traffic characteristics, and finally destroys the anonymous communication system to form an attack.

3.1.3. MSFA Attack Model. As shown in Figure 3, the attack model of MSFA scheme has been presented in this paper. When the user accesses the Internet through the ordinary browser or anonymous communication system, the original traffic information can be captured by Charles software or Wireshark [23] software.

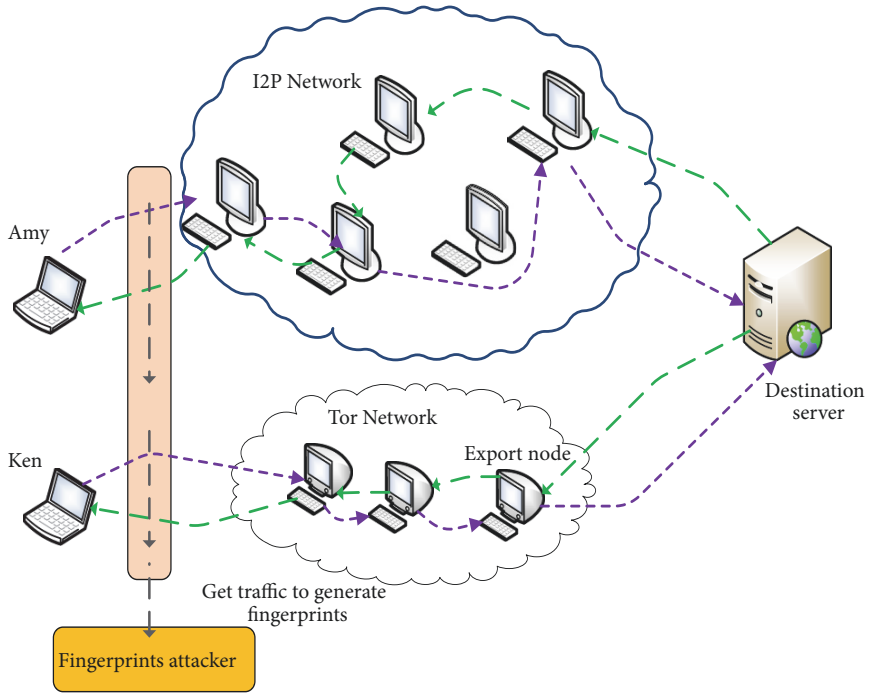


Figure 2: MSFA threat model.

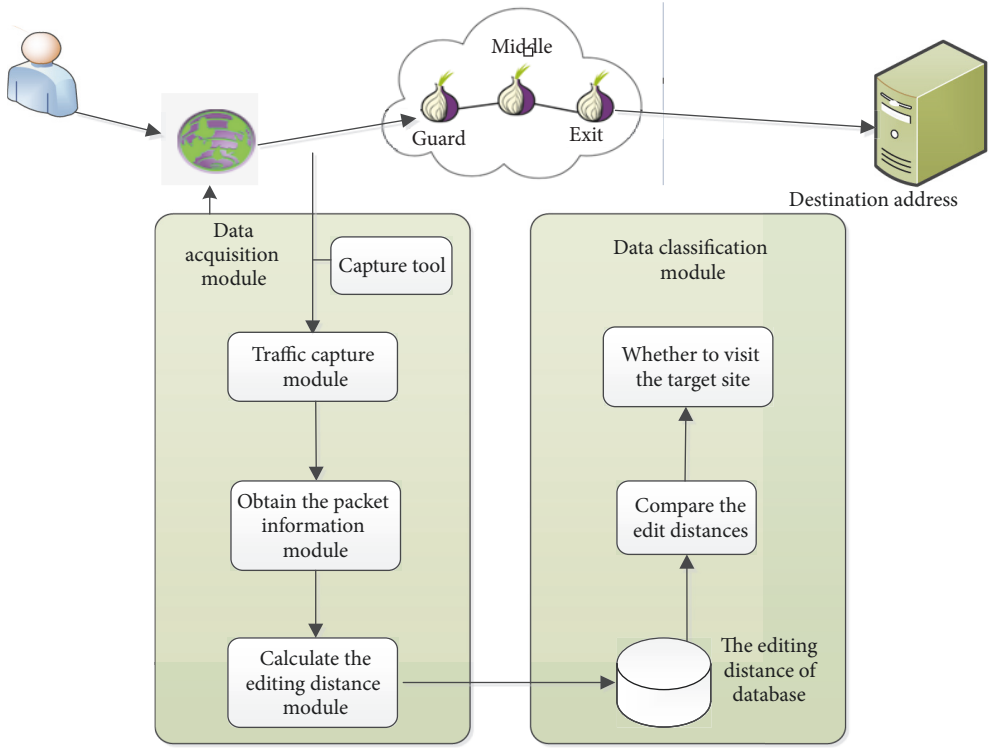


Figure 3: MASF attack model.

3.2. MSFA Scheme Module Design. We deeply study the module design of MSFA scheme (Figure 4), and mainly discuss the design of the traffic capture module, packet information acquisition module, and edit distance calculation module.

3.2.1. Traffic Capture Module Design. The experiment collects data from multiple target sites and organizes them according to the categories of target sites, giving each site a unique ID. In terms of data capture at the site, the experiment

Figure 4: Flow module design model.

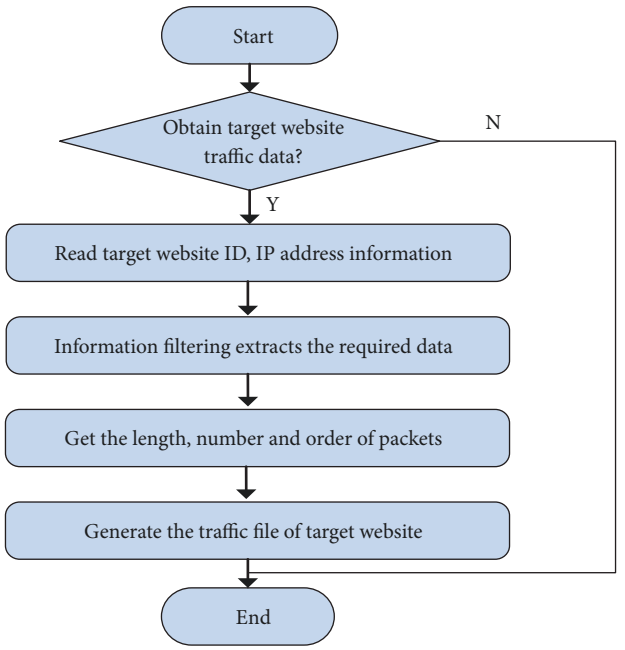


Figure 5: Get the packet information module design.

selects Wireshark to capture the traffic of TOR and I2P site.

3.2.2. Obtain the Packet Information Module Design. The module design for obtaining packet information is presented in Figure 5. In this experiment, we used Ubuntu system to capture the site traffic and Wireshark to capture data packets. The TCP/IP protocol defines the packets transmitted over the Internet, called IP Datagram. IP Datagram is independent of hardware and it consists of header and data. The first 20 bytes of the header are fixed length while the rear is an optional field whose length is variable. The source address and destination address of the header are both IP address.

The original data saved by Wireshark is complex while just the size and the number of Request and Response in the experiment need to be recorded to calculate the Damerau-Levenshtein distance. Therefore, the data needs to be processed. The experiment first groups the captured traffic data according to IP which must be the routing nodes of the Tor to obtain valuable packet information. In the experiment, after the traffic data is obtained, it only needs to request and respond to the packet size as it

```
(1) class Filter{
(2)   public:
        //Defines the storage server IP address; server_ips
(3)       unordered_set <string> server_ips;
        //Defines the storage client IP address: client_ips
(4)       unordered_set <string> client_ips;
        //Defines the maximum port that needs to be processed
(5)       int PROXYPORT_MIN;
        //Defines the minimum port that needs to be processed
(6)       int PROXYPORT_MAX;
(7)   public:
        //FilterConstructor
(8)       Filter(char* clientipfname, char* serveripfname, int portmin, int portmax);
        //Read the IP function
(9)       int read_ips(unordered_set<string>&set, char* fname);
        //Determines whether the file loads the function
(10)       bool is_onload(u_char* payload);
        //Determine whether the traffic function is listening
(11)       bool is_monitoredtraffic(char* src, unsigned int sport, char* dst, unsigned int dport);
        //Implement the transformation function for the data
(12)       RETparse_one(char* capfname, int proxy_port_min, intproxy_port_max, int remove_ack,
           char* monitoredoutname, char* localoutname, char* c2stau, char* s2ctau, char* timeseq);
(13)  };
```

Algorithm 1: Filter class code implementation.

```
(1) server_ips
(2) client_ips
(3) PROXYPORT_MIN
(4) PROXYPORT_MAX
(5) IF (Source address equals toclient_ips && Destination address equal toserver_ips
      && Source address port <= PROXYPORT_MAX
      && Source address port >= PROXYPORT_MIN
      && Destination address port <= PROXYPORT_MAX
      && Destination address port >= PROXYPORT_MIN)
(6) IF (Whether it is interrupted)
(7) IF (Destination address port <=PROXYPORT_MAX && Destination address port>=PROXYPORT_MIN)
(8)   Packet length* = -1
(9)   Output packet length
```

Algorithm 2: Algorithm for getting the packet information.

contains more complex data. Therefore, the original data needs to be processed. Filter class (Table 1) includes four attributes, namely, server_ips, client_ips, PROXYPORT_MIN, and PROXYPORT_MAX, and five methods, namely, Filter (), read_ips (), is_onload (), parse_one (), and is_monitoredtraffic (), and its partial details are shown in Algorithm 1. During the conversion of the website traffic file .cap, the lengths of the request and response packets are required in the experiment and saved in the file in turn. The request packet is identified as negative while the response packet is identified as positive. Algorithm 2 shows the algorithm for obtaining the packet information.

3.2.3. Calculate the Editing Distance Module Design. Get the length of the packet that the site accesses by obtaining the packet information module; then, it is necessary to calculate the editing distance between the two sites visit. In the experiment, the Damerau-Levenshtein distance, also known as the editing distance, is used which refers to the conversion of a string to another string in a minimum number of operations. The operation is defined as an insertion, deletion, or replacement of a character, or transposition of adjacent characters. In order to calculate the distance between the two sites, we use a matrix to store the distance and complete all edit distance calculations to output an edit distance matrix.

Table 1: Filter class functions and properties.

Filter	
+server_ips	Server-side IP
+client_ips	Client IP
+PROXYPORT_MIN	Minimum capture port
+PROXYPORT_	Maximum capture port
+filter()	Constructor
+read_ips()	Read the IP function
+is_onload()	Determines whether the file loads the function
+parse_one()	Conversion data function
+is_monitoredtraffic()	Determine the target traffic information function

Table 2: Calculate the Edit Distance Module algorithm.

Levenshtein	
+sizes	Store the length of file
+pool	Store the file pool of file
+str1	String 1
+str2	String 2
+websites	The number of storage sites
+trials	Store the number of visits per site
+distr	Store the distance
+get_distr	Get the distance function
+fetch_pool	Take the file function from the file pool
+is_size	To determine the length function
+read_size	Read the packet length function
+Parse_data	Conversion data function
+DLdis	Calculate the edit distance function

The algorithm for calculating the edit distance is shown in Table 2.

4. Implementation and Evaluation

4.1. Implementation of MSFA Scheme in the Tor Anonymous System Environment. The MSFA fingerprint attack scenario is tested under the Tor anonymous communication system.

4.1.1. Tor Anonymous System Installation and Configuration. In the experiment, we use the Linux system, download the corresponding Linux package on Tor official website, and then unpack the software package. Before running the Tor Browser, the global VPN proxy needs to be installed on the computer. Otherwise, the Tor Browser will not be able to establish a link and run normally.

4.1.2. Get the Packet Information. After the .pcap file is captured by the site, each .pcap file forms a .txt file that records the traffic information of the site.

4.2. MSFA Scheme Experimental Data Processing. The data processing of the MSFA scheme is discussed in detail, and the data of different kinds of websites are classified.

4.2.1. Calculate the Edit Distance. After fetching the packet length information required for the experiment, the next step is to further process the packet information and calculate the Damerau-Levenshtein edit distance. We still use the CAI fingerprint attack [CZJ2012] to standardize the edit distance to compensate for the changes of the packet tracking length. If d(t, t') means Damerau-Levenshtein edit distance, the fingerprint attack will normalize the edit distance as follows:

$$L\left(\mathrm{t}, t'\right) = \frac{d\left(t, t'\right)}{\min\left(|t|, \left|t'\right|\right)}. \tag{1}$$

$|t|$ represents the packet length in trace t, and the classifier normalizes the shortest value of two lengths. If the difference between t and t' is very large in length, then these two may come from different pages. In this case, dividing by $\min(|t|, |t'|)$ will result in a larger normalized distance, which is a feasible standardized distance. The implementation of calculation is shown in Algorithm 3.

4.2.2. Data Processing. According to the collection of the sites, we sort and select a few for processing. The specific process is as follows:

Select msn.com to do the experiment. The MSN was accessed through the Tor anonymous system at different times, with a total of 10 visits. The data for the site traffic was formed after the Wireshark captured the accessing traffic and the Filter class handled the file. The data is shown in Table 3.

After obtaining the traffic information for 10 visits to the msn.com website, we use Levenshtein_cantor_mpi to calculate the edit distance for this 10 traffic, as shown in Table 4.

When calculating the edit distance, the string that is accessed by the two traffic records of the site is compared. The smaller the edit distance is, the more similar the two records are.

By comparison, we find that the minimum distance of edit distance is 0.069 which is the fourth visit and the ninth visit msn.com site between the two edit distances. The maximum is 2.64 which is the editing distance between the first and fifth visiting. So we can initially determine that the edit distance of accessing MSN website ranges from 0 to 2.64. At the same time, the smallest distance to the other visiting average edit distance is selected as msn.com website fingerprint to store into the fingerprint database. By comparison, the minimum of average editing distance is between the second and other visiting msn.com, so the second visiting is put as a fingerprint of msn.com into the database.

Select Amazon to do the experiment. Amazon was accessed through the Tor anonymous system at different times, with a total of 10 visits. After Wireshark captured the accessing traffic and the Filter class handled the file, it formed the data for the site traffic. The data is shown in Table 5.

```
(1) double Levenshtein :: DLdis(int ms, int ns)
(2) {
(3)     double ret = 0;
(4)     int min;
        //Pretreatment
(5)     int m = ms;
(6)     int n = ns;
        //min takes the smaller between m and n
(7)      min = m < n ? m : n;
(8)      min = min = = 0 ? 1 : min;
(9)      int i, j;
(10)     double subcost, transcost;
        //Define operating costs to two
(11)     double idcost = 2;
        //Store the distance array
(12)     double** dis = new double*[m];
        //Initialize the array
(13)     for(i = 0;i < m;i++)
(14)         dis[i] = new double[n];
(15)     for(i = 0; i < m;i++)
(16)         for(j = 0; j < n; j++)
(17)             dis[i][j] = -1;
        //Calculate the operating costs of the first ramp line and the first vertical line
(18)     for(i = 0; i < m; i++)
(19)         dis[i][0] = i * idcost;
(20)     for(j = 0; j < n; j++)
(21)         dis[0][j] = j * idcost;
        //Calculate the operating costs of non-first rungs and first vertical lines.
(22)     for(i = 1; i < m; i++)
(23)     {
(24)         for(j = 1; j < n; j++)
(25)         {
                //If the two strings are equal, the operating cost is zero.
(26)     if(str1[i] = = str2[j])
(27)             subcost = transcost = 0;
(28)     else
(29)             {
                    //Otherwise the replacement cost is two.
(30)                 subcost = 2;
                    //The exchange cost is 0.1
(31)                transcost = 0.1;
(32)            }
                //The minimum cost is the edit distance, which is stored in the matrix.
(33)            dis[i][j] = minimum(dis[i-1][j] + idcost, dis[i][j-1] + idcost, dis[i-1][j-1] + subcost);
                //Two character exchanges
(34)     if(i >1 && j >1 && str1[i] = = str2[j-1] && str1[i-1] = = str2[j])
(35)             dis[i][j] = dis[i][j] < dis[i-2][j-2] + transcost ? dis[i][j]: dis[i-2][j-2] + transcost;
(36)     }
(37)  }
        //Free dis
(38)  for(i = 0; i < m; i++)
(39)      delete[] dis[i];
(40)      delete[] dis;
(41) }
```

Algorithm 3: Calculation of the edit distance implementation.

Table 3: The partial traffic of MSN site.

1	Upstream	-611	-68	-68	-68	-863	-1416	-68	-68	-611	-68	-68	-68	-863
	Downstream	68	611	611	68	611	1416	1416	1416	1416	1416	1416	1416	1416
2	Upstream	-68	-68	-1125	-68	-68	-68	-68	-68	-68	-68	-1416	-68	-863
	Downstream	68	611	611	68	611	68	611	68	611	1416	1348	68	611
3	Upstream	-611	-68	-68	-68	-252	-68	-68	-68	-1416	-1416	-68	-68	-68
	Downstream	611	68	68	68	611	1416	1416	1416	1416	1416	1416	1416	1416
4	Upstream	-611	-68	-68	-68	-805	-1416	-805	-1416	-68	-1125	-68	-68	-68
	Downstream	68	611	611	68	611	1416	1416	1416	1416	1416	1416	1416	1416
5	Upstream	-68	-68	-68	-611	-68	-1416	-1416	-68	-543	-68	-68	-1416	-1416
	Downstream	68	68	68	611	611	1416	1416	1416	1416	1416	1416	1416	1416
6	Upstream	-68	68	-68	-1154	-68	-68	-68	-68	-68	-68	-349	-68	-1416
	Downstream	68	611	611	68	611	1416	1416	1416	1416	1416	1416	1416	1416
7	Upstream	68	68	-68	1416	68	68	1416	68	68	68	68	68	68
	Downstream	68	611	611	611	68	611	68	611	611	68	611	1416	1416
8	Upstream	-68	-68	-68	-68	-68	-68	-1416	-68	-68	-68	-68	-68	-68
	Downstream	68	68	611	68	68	80	611	68	611	68	611	1416	805
9	Upstream	-68	-68	-1416	-68	-68	-68	-68	-68	-1406	-68	-68	-68	-1416
	Downstream	68	68	68	611	68	611	1416	1416	1416	1416	1416	1416	1416
10	Upstream	-68	-1416	-68	-68	-68	-1416	-1416	-68	-68	-68	-68	-1416	-1416
	Downstream	68	68	611	1416	291	68	1416	1416	1416	1416	1416	1416	1416

After obtaining the traffic information for 10 visits to the Amazon website, we use Levenshtein_cantor_mpi to calculate the edit distance for this 10 traffic, as shown in Table 6.

By comparison, we find that the minimum distance of edit distance is 0.034 which is the fourth visit and the fifth visit of amazon.com site between the two edit distances. The maximum is 11.81 which is the editing distance between the first and ninth visiting. So we can initially determine that the edit distance of accessing Amazon website ranges from 0 to 11.81. At the same time, the smallest distance to the other visiting average edit distance is selected as amazon.com website fingerprint to store into the fingerprint database. By comparison, the minimum of average editing distance is between the 9th and other visiting amazon.com, so the 9th visiting is put as a fingerprint of amazon.com into the database.

Select YouTube to do the experiment. YouTube was accessed through the Tor anonymous system at different times, with a total of 10 visits. After Wireshark captured the accessing traffic and the Filter class handled the file, it formed the data for the site traffic. The data is shown in Table 7.

The Levenshtein_cantor_mpi is used to calculate the edit distance for the 10 traffic which has been obtained by 10 visits to the youtube.com website, as shown in Table 8.

By comparison, we find that the minimum distance of edit distance is 0.27 which is the 8th visit and the 7th visit of youtube.com site between the two edit distances. The maximum is 8.44 which is the editing distance between the 10th and 9th visiting. So we can initially determine that the edit distance of accessing YouTube website ranges from 0 to 8.44. At the same time, the smallest distance to the other visiting average edit distance is selected as youtube.com website fingerprint to store into the fingerprint database. By comparison, the minimum of average editing distance is between the 8th and other visiting youtube.com, so the 8th visiting as a fingerprint of youtube.com is put into the database.

5. Conclusion

The real spreading of IoT services requires customized security and privacy levels to be guaranteed. Many IoT services and applications may expose sensitive and personal information which may be abused by attackers. As such, privacy protection must be considered and it is a core requirement in any IoT ecosystem. The MSFA attack scheme proposed in this paper is based on the edit distance to compare the similarity between the two visits. Firstly, the differences between the different types of website traffic can be observed from the above data. Take the Amazon which is the e-commerce website and YouTube which is the video website as examples. The traffic of Amazon website ranges from 0 to 11.81, while the traffic of YouTube video ranges from 0 to 8.44. Secondly, the length of the traffic data has an impact on the edit distance. In general, the longer the length of the site traffic data, the greater the edit distance of the site access traffic and the larger the range.

Data Availability

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

Table 4: The edit distance of MSN accessing traffic between each other.

	1	2	3	4	5	6	7	8	9	10
1	0	0.78	0.954065	0.083248	2.64735	0.869404	1.300171	1.215214	2.223183	0.948547
2	0.78	0	0.553862	1.591451	0.885303	1.047593	0.667157	0.787618	0.246367	1.110267
3	0.954065	0.553862	0	0.241667	0.545528	1.403252	1.972764	1.806911	1.641869	1.489634
4	0.083248	1.591451	0.241667	0	1.065391	0.869876	2.082786	2.279481	0.069204	0.867333
5	2.64735	0.885303	0.545528	1.065391	0	1.719807	1.169649	1.345401	0.0609	1.74
6	0.869404	1.047593	1.403252	0.869876	1.719807	0	0.64718	0.641953	0.150519	0.603576
7	1.300171	0.667157	1.972764	2.082786	1.169649	0.64718	0	0.544458	0.228374	0.772
8	1.215214	0.787618	1.806911	2.279481	1.345401	0.641953	0.544458	0	0.958478	0.7032
9	2.223183	0.246367	1.641869	0.069204	0.0609	0.150519	0.228374	0.958478	0	0.289273
10	0.948547	1.110267	1.489634	0.867333	1.74	0.603576	0.772	0.7032	0.289273	0
	1.102118	0.766962	1.060955	0.915044	1.117933	0.795316	0.938454	1.028271	0.586817	0.852383

Table 5: The partial traffic of Amazon site.

1	Upstream	-68	-68	-68	-68	-68	-68	-68	-68	-1416	-68	-68	-68	-68
	Downstream	68	68	611	68	611	68	611	68	611	1416	1416	1416	1416
2	Upstream	-68	-68	-68	-68	-68	-68	-68	-68	-1026	-68	-68	-68	-68
	Downstream	68	611	611	6868	611	1416	1416	999	1416	999	1416	805	68
3	Upstream	-611	-68	-611	-68	-611	-68	-1416	-1416	-1416	-68	-68	-611	-611
	Downstream	68	611	1416	1416	1416	1416	68	611	68	1416	291	1416	1416
4	Upstream	-68	-68	-68	-1416	-68	-68	-68	-68	-68	-68	-68	-68	-68
	Downstream	68	-611	611	1125	611	1125	611	1416	1416	1416	1416	68	1416
5	Upstream	-68	-863	-68	-68	-68	-68	-68	-68	-68	-1416	-611	-68	-68
	Downstream	1125	68	1125	1416	1416	1416	1416	1416	1416	1416	1416	1416	1416
6	Upstream	-68	-68	-68	-611	-68	-611	-68	-291	-1416	-68	-68	-572	-68
	Downstream	68	68	68	611	68	1416	1416	1416	1416	1416	1416	1416	1416
7	Upstream	-1416	-1416	-68	-68	-1416	-1416	-805	-1416	-1416	-80	-1416	-1416	-1416
	Downstream	68	68	68	611	1416	1416	1416	1416	1416	1416	1416	1416	1416
8	Upstream	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68
	Downstream	68	611	611	68	611	68	611	1416	1416	1416	1416	388	68
9	Upstream	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68	-68
	Downstream	68	68	611	1416	1416	1416	1416	68	1416	1416	1416	514	1416
10	Upstream	-68	-68	-68	-80	-68	-68	-68	-68	-68	-68	-68	-68	-68
	Downstream	68	-611	611	68	611	1125	611	1416	1416	1416	1416	68	611

Table 6: The edit distance of Amazon accessing traffic between each other.

	1	2	3	4	5	6	7	8	9	10
1	0	0.359215	0.320107	4.281042	4.388232	8.531568	5.133218	11.81551	4.401792	0.00311
2	0.359215	0	0.223082	3.543489	3.637893	7.273014	4.296347	10.16845	3.64932	0.357429
3	0.320107	0.223082	0	3.69364	3.790635	7.539715	4.453549	10.16845	3.807355	0.319185
4	4.281042	3.543489	3.69364	0	0.034504	1.460591	0.431633	2.534893	0.267367	4.276196
5	4.388232	3.637893	3.790635	0.034504	0	1.405703	0.397243	2.462701	0.272188	4.383303
6	8.531568	7.273014	7.539715	1.460591	1.405703	0	1.063288	0.842246	1.407332	8.519348
7	5.133218	4.296347	4.453549	0.431633	0.397243	1.063288	0	2.008422	0.395934	5.124948
8	11.81551	10.16845	10.16845	2.534893	2.462701	0.842246	2.008422	0	2.452807	11.79957
9	4.401792	3.64932	3.807355	0.267367	0.272188	1.407332	0.395934	2.452807	0	4.395612
10	0.00311	0.357429	0.319185	4.276196	4.383303	8.519348	5.124948	11.79957	4.395612	0
	3.923379	3.350824	3.465262	2.052336	2.07724	3.804276	2.330453	5.458984	2.104971	3.91786

Table 7: The partial traffic of YouTube site.

1	Upstream	-1109	-1109	-595	-595	-1109	-595	-1109	-595	-1109	-1109	-595	-1400	-1332
	Downstream	595	595	595	595	595	1400	1400	1400	692	1400	304	1400	1400
2	Upstream	595	-595	-595	-1138	-595	-1109	-595	-1400	-1400	-527	-1138	-1400	-595
	Downstream	595	1400	1400	983	1109	595	595	1109	1400	1400	1400	275	595
3	Upstream	-1109	-595	-595	-595	-595	-1109	-1109	-595	-595	-595	-595	-1109	-1138
	Downstream	595	1400	1400	1400	721	595	595	595	595	595	595	1400	1400
4	Upstream	-595	-595	-595	-1109	-595	-595	-595	-595	-595	-595	-595	-1109	-1138
	Downstream	595	595	595	595	1400	1400	1400	692	595	595	595	1109	595
5	Upstream	-1109	-1109	-595	-1400	-847	-195	-1138	-1109	-1109	-1400	-304	-1400	-595
	Downstream	595	1400	1400	1400	1400	1400	459	1400	1400	1400	178	595	595
6	Upstream	-595	-595	-595	-1109	-1109	-595	-595	-1109	-595	-1109	-595	-595	-595
	Downstream	595	595	595	595	595	595	692	595	595	1109	595	595	595
7	Upstream	-595	-595	-595	-1109	-595	-1106	-595	-1109	-595	-595	-595	-1400	-333
	Downstream	595	595	595	595	1400	1400	1400	692	595	595	1109	595	595
8	Upstream	-595	-595	-595	-595	-1109	-595	-1109	-595	595	-595	-595	-1109	-1109
	Downstream	595	595	595	595	1400	1400	1400	692	595	1400	304	595	595
9	Upstream	-595	-595	-595	-1109	-595	-1109	-595	-595	-595	-595	-595	-595	-1109
	Downstream	595	595	595	595	1400	1400	1400	692	595	595	1109	595	1109
10	Upstream	-1109	-1109	-595	-1138	-595	-1400	-1399	-1400	-1390	-1400	-1400	-1388	-1400
	Downstream	595	1400	1400	1400	1400	1400	1400	1400	1400	595	1400	1400	595

Table 8: The edit distance of YouTube accessing traffic between each other.

	1	2	3	4	5	6	7	8	9	10
1	0	0.301	1.371	0.713	4.953	2.456	0.632	0.659	0.895	5.584
2	0.301	0	1.43	0.695	5.069	2.528	0.683	0.705	0.871	5.725
3	1.371	1.43	0	2.307	2.389	0.879	0.754	0.7	2.61	2.801
4	0.713	0.695	2.307	0	6.953	3.706	1.333	1.381	0.145	7.747
5	4.953	5.069	2.389	6.953	0	1.4224	3.677	3.526	7.581	0.574
6	2.456	2.528	0.879	3.706	1.424	0	1.644	1.572	4.107	1.748
7	0.632	0.683	0.754	1.333	3.677	1.644	0	0.27	1.562	4.194
8	0.659	0.705	0.7	1.381	3.526	1.572	0.27	0	1.614	4.041
9	0.895	0.871	2.61	0.145	7.581	4.107	1.562	1.614	0	8.44
10	5.584	5.725	2.801	7.747	0.574	1.748	4.194	4.041	8.44	0
	1.756	1.801	1.542	2.498	3.615	2.006	1.475	1.447	2.782	4.085

Conflicts of Interest

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

Acknowledgments

This work is supported by the following grants: the National Natural Science Foundation of China under Grant no. 61170273; the China Scholarship Council under Grant no. [2013]3050. We thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] R.-H. Hsu, J. Lee, T. Q. S. Quek, and J.-C. Chen, "GRAAD: group anonymous and accountable D2D communication in mobile networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 449–464, 2018.

[2] R. Amin, S. Islam, G. Biswas, M. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 80, pp. 483–494, 2015.

[3] M. Xie, M. Huang, Y. Bai, and Z. Hu, "The anonymization protection algorithm based on fuzzy clustering for the ego of data in the internet of things," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 2970673, 10 pages, 2017.

[4] S. Sciancalepore, G. Oligeri, G. Piro, G. Boggia, and R. Di Pietro, "EXCHANge: Securing IoT via channel anonymity," *Computer Communications*, vol. 134, pp. 14–29, 2019.

[5] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 23rd USENIX Security Symposium*, USENIX Association, San Diego, CA, USA, 2014.

[6] O. Berthold, H. Federrath, and S. Köpsell, "Web MIXes: a system for anonymous and unobservable internet access," in *Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, vol. 2009 of *Lecture Notes in Computer Science*, pp. 115–129, Springer, 2000.

[7] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in *Proceedings of International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, vol. 2009 of *Lecture Notes in Computer Science*, pp. 46–66, Springer, 2000.

[8] B. Zantout and R. Haraty, "I2P data communication system," in *Proceedings of the 10th International Conference on Networks*, pp. 401–409, 2011.

[9] J. Al-Muhtadi, M. Qiang, K. Saleem, M. AlMusallam, and J. J. Rodrigues, "Misty clouds—A layered cloud platform for online user anonymity in Social Internet of Things," *Future Generation Computer Systems*, vol. 92, pp. 812–820, 2019.

[10] A. Das, N. Borisov, and E. Chou, "Every Move You Make: Exploring Practical Issues in Smartphone Motion Sensor Fingerprinting and Countermeasures," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 1, pp. 88–108, 2018.

[11] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: Website fingerprinting attacks and defenses," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012*, pp. 605–616, October 2012.

[12] T. Wang and I. Goldberg, "Improved website fingerprinting on Tor," in *Proceedings of the the 12th ACM workshop*, pp. 201–212, Berlin, Germany, November 2013.

[13] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the 21st ACM Conference on Computer and Communications Security, CCS 2014*, pp. 263–274, November 2014.

[14] R. Nithyanand, X. Cai, and R. Johnson, "Glove: a bespoke website fingerprinting defense," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES 2014, in Conjunction with the ACM Conference on Computer and Communications Security, ACM CCS 2014*, pp. 131–134, 2014.

[15] A. Kwon, M. AlSabah, D. Lazar et al., "Circuit fingerprinting attacks: passive deanonymization of Tor hidden services," in *Proceedings of the 24th USENIX Security Symposium*, pp. 287–301, 2015.

[16] T. Wang and I. Goldberg, "On realistically attacking Tor with website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 21–36, 2016.

[17] R. Overdorf, M. Juarez, G. Acar, R. Greenstadt, and C. Diaz, "How unique is your .onion? An analysis of the fingerprintability of tor onion services," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2021–2036, USA, November 2017.

[18] G. Cherubin, J. Hayes, and M. Juarez, "Website Fingerprinting Defenses at the Application Layer," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 186–203, 2017.

[19] X. Luo, P. Zhou, E. W. W. Chan et al., "HTTPOS: Sealing information leaks with browser-side obfuscation of encrypted flows," in *Proceedings of the 2011 Network and Distributed System Security Symposium*, San Diego, CA, USA, 2017.

[20] E. Brill and C. R. Moore, "An improved error model for noisy channel spelling correction," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL '00)*, pp. 286–293, October 2000.

[21] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, 1964.

[22] M. Li, M. Zhu, Y. Zhang, and M. Zhou, "Exploring distributional similarity based models for query spelling correction," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, COLING/ACL 2006*, pp. 1025–1032, Australia, July 2006.

[23] Wireshark, "Wireshark network protocol analyzer," 2017, http://www.wireshark.org/.