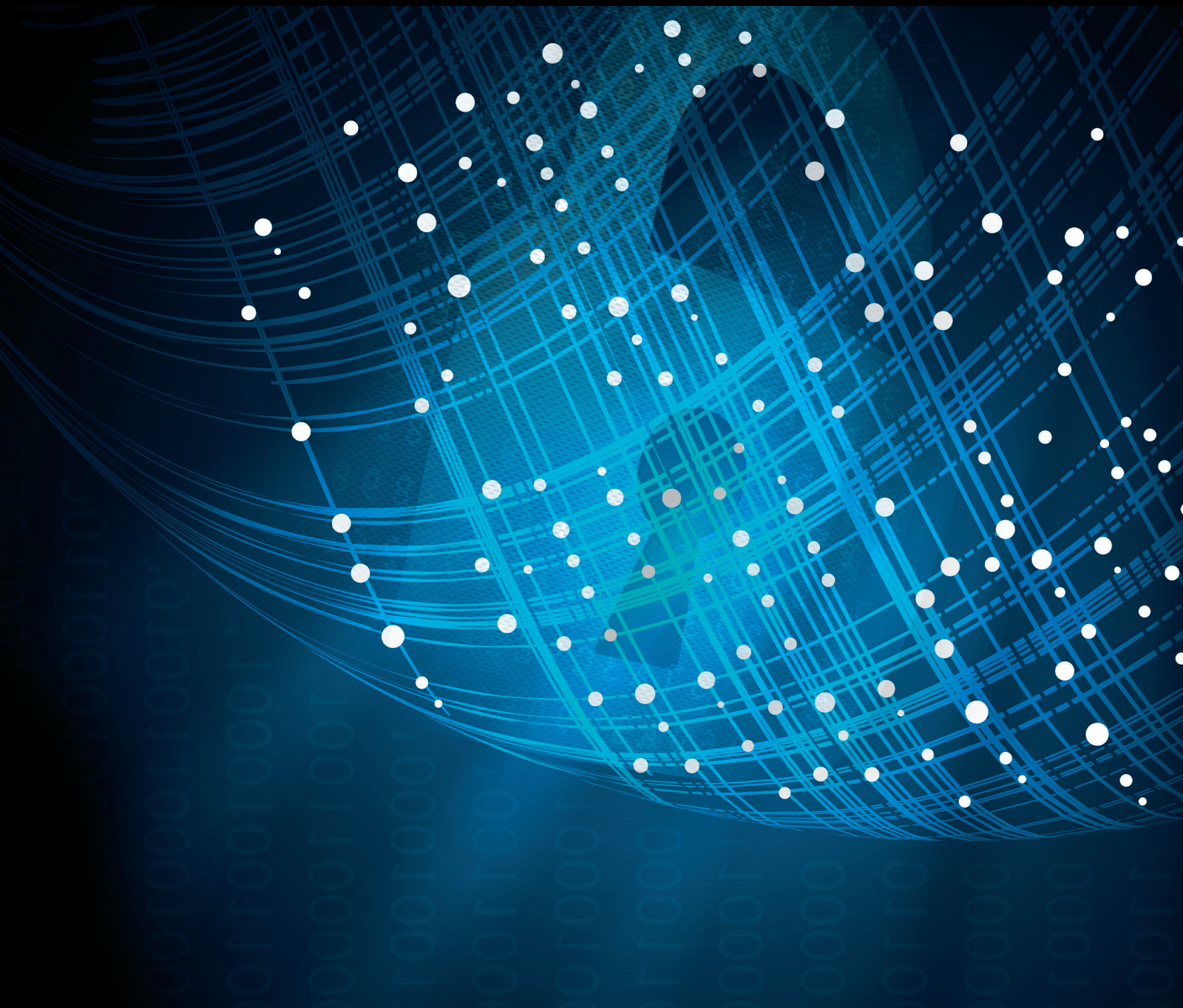


Security and Communication Networks

Cyberspace Security for Future Internet

Lead Guest Editor: Dafang Zhang

Guest Editors: Guojun Wang, Xin Wang, Zhenyu Li, and Wenjia Li





Cyberspace Security for Future Internet

Security and Communication Networks

Cyberspace Security for Future Internet

Lead Guest Editor: Dafang Zhang

Guest Editors: Guojun Wang, Xin Wang, Zhenyu Li,
and Wenjia Li



Copyright © 2018 Hindawi. All rights reserved.

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

Editorial Board


Mamoun Alazab, Australia
Cristina Alcaraz, Spain
Angelos Antonopoulos, Spain
Frederik Armknecht, Germany
Benjamin Aziz, UK
Alessandro Barengi, Italy
Pablo Garcia Bringas, Spain
Michele Bugliesi, Italy
Pino Caballero-Gil, Spain
Tom Chen, UK
Alessandro Cilardo, Italy
Stelvio Cimato, Italy
Vincenzo Conti, Italy
Salvatore D'Antonio, Italy
Paolo D'Arco, Italy
Alfredo De Santis, Italy
Angel M. Del Rey, Spain
Roberto Di Pietro, France
Jesús Díaz-Verdejo, Spain
Nicola Dragoni, Denmark
Carmen Fernandez-Gago, Spain
Clemente Galdi, Italy
Dimitrios Geneiataakis, Italy

Bela Genge, Romania
Debasis Giri, India
Francesco Gringoli, Italy
Jiankun Hu, Australia
Ray Huang, Taiwan
Tao Jiang, China
Minho Jo, Republic of Korea
Bruce M. Kapron, Canada
Kiseon Kim, Republic of Korea
Sanjeev Kumar, USA
Maryline Laurent, France
Huaizhi Li, USA
Zhe Liu, Canada
Pascal Lorenz, France
Leandros Maglaras, UK
Emanuele Maiorana, Italy
Vincente Martin, Spain
Fabio Martinelli, Italy
Barbara Masucci, Italy
Jimson Mathew, UK
David Megias, Spain
Leonardo Mostarda, Italy
Qiang Ni, UK


Petros Nicopolitidis, Greece
David Nuñez, USA
A. Peinado, Spain
Gerardo Pelosi, Italy
Gregorio Martinez Perez, Spain
Pedro Peris-Lopez, Spain
Kai Rannenberg, Germany
Francesco Regazzoni, Switzerland
Khaled Salah, UAE
Salvatore Sorce, Italy
Angelo Spognardi, Italy
Sana Ullah, Saudi Arabia
Ivan Visconti, Italy
Guojun Wang, China
Zheng Yan, China
Qing Yang, USA
Kuo-Hui Yeh, Taiwan
Sherali Zeadally, USA
Zonghua Zhang, France
Kim-Kwang Raymond Choo, USA
Jong-Hyook Lee, Republic of Korea

Contents

Cyberspace Security for Future Internet

Dafang Zhang , Guojun Wang, Xin Wang, Zhengyu Li, and Wenjia Li
Editorial (1 page), Article ID 5313980, Volume 2018 (2018)


Session Fingerprinting in Android via Web-to-App Intercommunication

Efthimios Alepis and Constantinos Patsakis 
Research Article (13 pages), Article ID 7352030, Volume 2018 (2018)




Towards a New Algorithm to Optimize IPv6 Neighbor Discovery Security for Small Objects Networks

Ali El Ksimi  and Cherkaoui Leghris
Research Article (11 pages), Article ID 1816462, Volume 2018 (2018)

A Feasible Fuzzy-Extended Attribute-Based Access Control Technique

Yang Xu , Wuqiang Gao, Quanrun Zeng, Guojun Wang, Ju Ren, and Yaoxue Zhang
Research Article (11 pages), Article ID 6476315, Volume 2018 (2018)

Demadroid: Object Reference Graph-Based Malware Detection in Android

Huanran Wang , Hui He , and Weizhe Zhang 
Research Article (16 pages), Article ID 7064131, Volume 2018 (2018)


BAVP: Blockchain-Based Access Verification Protocol in LEO Constellation Using IBE Keys

Songjie Wei , Shuai Li , Peilong Liu , and Meilin Liu 
Research Article (14 pages), Article ID 7202806, Volume 2018 (2018)

An Efficient and Privacy-Preserving Multiuser Cloud-Based LBS Query Scheme

Lu Ou , Hui Yin , Zheng Qin , Sheng Xiao , Guangyi Yang, and Yupeng Hu
Research Article (11 pages), Article ID 4724815, Volume 2018 (2018)

Quantum Cryptography for the Future Internet and the Security Analysis

Tianqi Zhou, Jian Shen , Xiong Li, Chen Wang, and Jun Shen
Research Article (7 pages), Article ID 8214619, Volume 2018 (2018)

Editorial

Cyberspace Security for Future Internet

Dafang Zhang ¹, **Guojun Wang**,² **Xin Wang**,³ **Zhengyu Li**,⁴ and **Wenjia Li**⁵

¹Hunan University, Changsha 410082, China

²Guangzhou University, Guangzhou 510006, China

³Stony Brook University, Stony Brook, NY 11794-2350, USA

⁴Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing 100190, China

⁵New York Institute of Technology, New York, 10023, USA

Correspondence should be addressed to Dafang Zhang; liyanbiao@ict.ac.cn

Received 10 July 2018; Accepted 10 July 2018; Published 5 August 2018

Copyright © 2018 Dafang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cyberspace is the most popular environment for information exchange whose security suffers from ever-increasing challenges with the rapid development of the Internet. This issue published 7 latest contributions on cyberspace security for future Internet.

Due to the fast advance of mobile technologies and mobile applications, smartphones, especially Android devices, are widely used in our daily life, which are threatened greatly by attacks. Therefore, malware analysis on Android platform is in urgent demand. Regarding drawbacks of existing static and dynamic analysis approaches, a new framework is introduced here, which can better satisfy the demand for actual use. In addition to malware detection, privacy is another big concern. For example, mobile devices are always equipped with numerous sensors, which may reveal sensitive information when correlated with other data or sources. How to protect user privacy or identify privacy risks exposed by applications? Some novel user deanonymization approaches and user fingerprinting in Android are introduced. Besides, location-based services (LBSs) become more and more popular in mobile Internet, such as map directions, restaurant recommendations, and taxi reservations. Regarding the privacy of personal location information, an efficient and privacy-preserving multiuser query scheme is presented for cloud-based SBSs.

Access control, on the other hand, also plays a very important role in cyberspace security. Regarding the absence of a flexible exceptional approval mechanism in attribute-based access control (ABAC), a feasible fuzzy-extended ABAC

technique is presented, which improves the flexibility in urgent exceptional authorizations and thereby improves the resource usability and business timeliness. In the field of satellite communication, existing centralized authentication protocols for MEO/GEO satellite networks cannot accommodate LEO satellite networks with frequent user connection switching. Combining identity-based encryption and the block-chain technology, a fast and efficient access verification protocol is introduced.

New architectures and new computing technologies bring in both opportunities and challenges in cyberspace security. In Small Object Networks with IPv6, the process of Duplicate Address Detection is subject to many attacks. In view of this, a new algorithm to optimize the security in IPv6-DAD is presented. With quantum computers, cyberspace security has become the most critical issue in the Internet in near future. So, characteristics of the quantum cryptography and how to use it in future Internet are analyzed.

Dafang Zhang
Guojun Wang
Xin Wang
Zhengyu Li
Wenjia Li

Research Article

Session Fingerprinting in Android via Web-to-App Intercommunication

Efthimios Alepis and Constantinos Patsakis 

Department of Informatics, University of Piraeus, 80 Karaoli & Dimitriou, 18534 Piraeus, Greece

Correspondence should be addressed to Constantinos Patsakis; kpatsak@gmail.com

Received 29 December 2017; Accepted 3 June 2018; Published 28 June 2018

Academic Editor: Guojun Wang

Copyright © 2018 Efthimios Alepis and Constantinos Patsakis. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The extensive adoption of mobile devices in our everyday lives, apart from facilitating us through their various enhanced capabilities, has also raised serious privacy concerns. While mobile devices are equipped with numerous sensors which offer context-awareness to their installed apps, they can also be exploited to reveal sensitive information when correlated with other data or sources. Companies have introduced a plethora of privacy invasive methods to harvest users' personal data for profiling and monetizing purposes. Nonetheless, up till now, these methods were constrained by the environment they operate, e.g., browser versus mobile app, and since only a handful of businesses have actual access to both of these environments, the conceivable risks could be calculated and the involved enterprises could be somehow monitored and regulated. This work introduces some novel user deanonymization approaches for device and user fingerprinting in Android. Having Android AOSP as our baseline, we prove that web pages, by using several inherent mechanisms, can cooperate with installed mobile apps to identify which sessions operate in specific devices and consequently further expose users' privacy.

1. Introduction

The unprecedented growth of mobile usage has radically transformed our daily lives. In addition to the great advances in our communications, mobile devices have changed the way we create, process, and consume information, as they realize pervasive and ubiquitous computing. Among others, one of the most significant emerged changes is how we value information. The fact that people are constantly and effortlessly connected to the Internet via smart devices which empower people's unobstructed communication, information flow, and entertainment in many occasions results in disregarding or underestimating the value of the information they consume and offer to third parties. This kind of collected data is considered as the world's new oil [1] but is also accompanied by an increased risk regarding users' privacy.

Subsequently, as far as information offering is concerned, the value of the provided information to third parties is in most of the cases considerably high, something that is not always understood by the users. For instance, one might share his location with an app or a web page neglecting the fact that

this single piece of information also encloses a very sensitive piece of data which can be exploited for various purposes. Indicative uses for such location sharing could be the recommendation of other users in proximity for communication purposes or even for sharing a ride. Aggregating location data from numerous users can provide real-time traffic analytics or insight into resource requirements in a smart city. Apparently, this information can stimulate businesses' prosperity by enabling the implementation of further customer-centered services. Therefore most companies are striving to extract as much information as possible from users.

While data mining offers undeniable advantages to users, e.g., service personalization can be considered as a noble cause, companies tend to exploit data even further for profiling and targeted advertising. Such tactics can expose users to many privacy hazards. This trend is highlighted by the fact that many companies are providing APIs which harvest user data to create fine-grained user profiles, containing a lot of sensitive user information. Such practices have also led to the introduction of methods such as browser and device fingerprinting. Nonetheless, mobile apps and web pages are

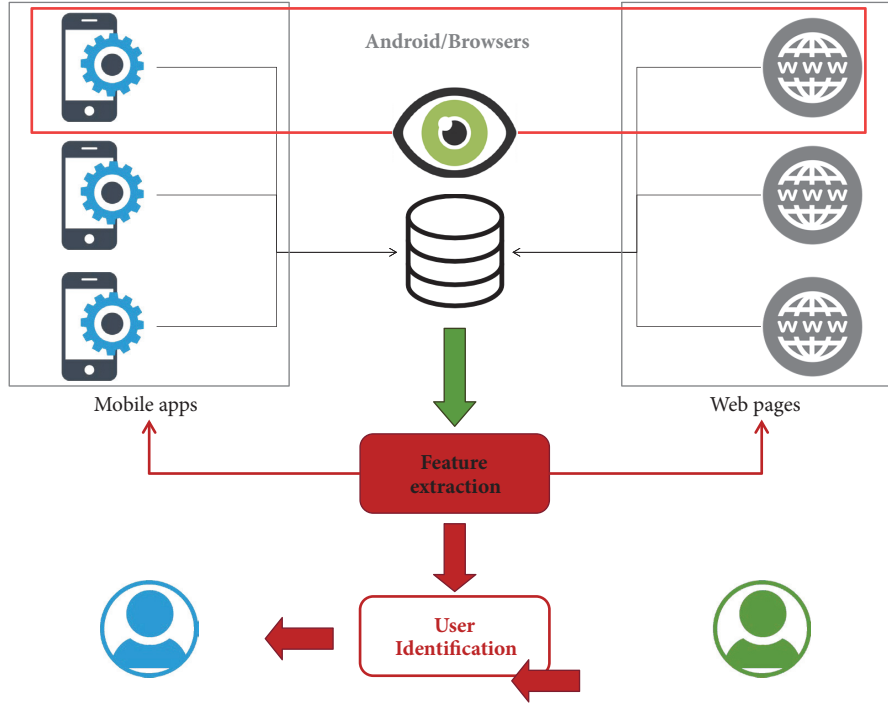


FIGURE 1: Basic concept.

thus far considered as two diverse ecosystems, as they refer to two discrete software environments with radical differences in their information flow and data usage. This distinction works in favor of users' privacy, since it allows some parts of their activities to remain isolated and hence private. For instance, it prevents an app from knowing which web pages a user visits or a web page from knowing which apps a user is using and when. On the contrary, enabling access between these two environments could allow for a web page to communicate with an installed app to recover further sensitive personal information from local files or sensor measurements and hence further reveal one's interests.

The goal of this work is to illustrate that there are currently several means to realize user identification in Android, regardless of the environment a software module is operating on. Despite the privacy hesitations that people might have towards the well-known tech giants or independent browsers, we provide some concrete examples proving that an "All Seeing Eye", a software entity able to monitor users' actions across both the web and the application ecosystems, can be easily created. Such an entity, in the form of an cloud-based database equipped with some additional services can correlate information from web pages and mobile apps in order to identify individuals. After a thorough investigation in the related scientific literature and to the best of our knowledge, the authors of this paper have concluded that this problem has been so far only partially studied, as current literature is focused on methods which examine each software ecosystem independently and not both of them as a whole. In fact, the proposed methods in this work can be considered as an extension of device fingerprinting as they

do not solely depend upon unique characteristics of device components or hardware identifiers. We label these methods as "session fingerprinting" since their goal is to reveal whether web-browsing and software sessions operate simultaneously in a device and identify the user.

The generic concept of this work, in a simplified form, is illustrated in Figure 1. Each side of this figure is dedicated to the two software "ecosystems", namely web pages and mobile apps. Obviously, there is a crosscut from the OS, namely, Android, since it manages calls from both ecosystems in a mobile device, as well as from the browsers which by definition belong as applications to both ecosystems. The "All Seeing Eye" acts as a Command and Control, C&C, server which collects information from web pages and apps, correlates it, and transmits "commands" and the corresponding information to both sides. The commands may range from "retrieve a list of installed apps" and "scan local storage for files containing X", to "display ad Y" or "application Z send webpage data W". Therefore, the "All Seeing Eye", as the orchestrator of all performed actions by apps and web pages can ultimately reveal user identities.

While similar attempts have been made in the past, it is rather important to note that methods trying to escape the browser's environment without users' consent are considered to be malware and usually exploit browsers' vulnerabilities. Especially in the case of Android, passing a single bit of information from a benign browser to an app is rather difficult, given the fact that it has to bypass not only the browser's sandbox but also additional obstacles due to Android's inherent security model which will be discussed later on.

This paper extends previous work of the authors [2] by providing more details for the underlying methods, the related literature, and also experiments regarding session fingerprinting. The rest of this work is organized as follows. In the next section we present the related work, discussing methods for user profiling in mobile devices and browsers and some Android specific details regarding permissions of apps. In Section 3, our newly introduced concept of “session fingerprinting” is analyzed and in Section 4 we state the problem we address and discuss how both apps and web pages are expected to behave in this context. In Section 5 we present four concrete examples which prove the efficacy of our approach and detail how they can be realized. Section 6 illustrates the extension of the threat by providing experimental result and statistics. Finally, we conclude discussing some of our findings and ideas for future research.

2. Related Work

2.1. Isolation of Apps in Android. Android started as a heavily modified Linux distribution to meet the needs of mobile devices which had significantly fewer resources than desktop computers. However, the introduced changes made it quite unique, leading many people to consider it something beyond a different Linux distribution.

Contrary to most operating systems, the actual user of the device does not have administrative privileges by default. While this choice is actually preventing the user to have complete control of the device he owns, it also prevents adversaries to gain more privileges than they should. Certainly, there are several attacks presented in the literature [3, 4]; however, they can be considered as few and quickly patched by Google. Since users tend to install a significant number of apps in Android, each application needs to have different access to the device resources in order to prevent security and privacy hazards. To this end, each Android app is a different OS user, to which the user, owner of the device, grants different permissions. By isolating each app, Android guarantees the integrity of the contents of each procedure and prevents other apps from accessing them. Moreover, since the user selects which apps are allowed to access specific resources, the user is able to control the information flow in his device. The latter was significantly improved in Android Marshmallow, as Google decided to introduce the runtime-permission model so that users can grant and revoke app permissions on “dangerous” resources, the ones that present the biggest privacy risk, for instance camera, microphone, and location. See Figure 2 for the full list of the so-called “dangerous permissions”.

To enforce the permission model Android has to perform several steps. Before describing this specific mechanism, it has to be highlighted that since each application in Android is considered as a different user, it is assigned a different UID. This prevents applications from accessing the data and private resources of the other installed apps, providing more security and privacy to Android. Each call to Android framework API is accompanied by the corresponding UID

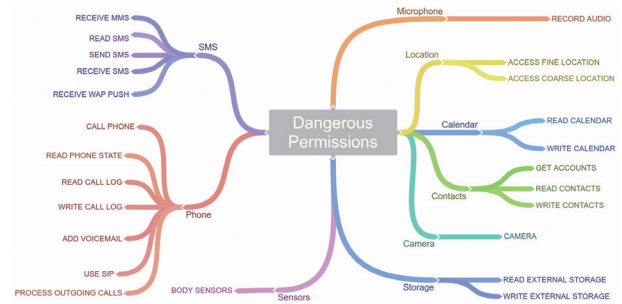


FIGURE 2: Dangerous permissions in Android [5].

of the app performing the call. Android checks whether the permission for the call has been assigned upon installation in the AndroidManifest.xml file and if this is the case, Android checks the permission level of this call (normal, dangerous, etc.). Normal permissions are automatically granted and access to the API is provided instantly. However, if the call is for a dangerous permission, the system will query whether access to this resource has been granted by the user during runtime and allow or deny the access accordingly. Finally, if the permission is signature or signatureOrSystem, then it is granted only to applications that are in the Android system image or that are signed with the same certificate as the application that declared the permission.

While this model may seem secure it does not prevent privacy exposure. The fact that apps have unrestricted access to the Internet allows them to communicate a lot of information. The latter is augmented by the fact that apps can profile their users with normal permissions as they may know, e.g., the apps that are installed, the WiFi networks a user has stored and is using, created and joined arbitrary networks, or even the users' whereabouts with features like WiFi P2P [6]. In general, while an app may use only normal permissions, this does not necessarily mean that it is benign [4, 5, 7]. An adversary model for exfiltrating data from Android devices has been studied in [8] as the use cases are numerous especially in an era when phones are shipped with numerous bloatware [9, 10].

SystemOfSignature permission allows Android apps to be granted a permission as long as an app with the same signature is granted this permission. Extending this concept, Davi et al. [11] showed that apps could escalate their access privileges by performing calls to other applications which had already been granted the privileges they wanted. Orthacker et al. [12] further extended the aforementioned scenario to show that an adversary could use *permission spreading*, that is, split the necessary privileges to different applications, and launch the attack through intercommunication. Similar approaches with app collusion and spread of permissions have been reported in the literature [13–15]; therefore researchers have been gradually focusing on more thorough analysis of intents [16].

In most of the Android cases documented by researchers, information is leaked from one app to another through a covert channel [17, 18]. Although Rushanan et al. in [19]

achieve a goal similar to ours, their study concerns only the desktop environment. Their approach consists in exploiting the Web Workers API in order to increase the CPU and memory utilization. By monitoring both CPU and memory usage, they manage to pass messages from a web page to an app in a desktop computer. Nevertheless, this attack scenario is not possible in an Android device. For devices up to Marshmallow, while apps could monitor the `/proc/` directory and extract some information about memory usage, the recovered information is far from being considered fine-grained and does not include CPU usage. With the introduction of Nougat, apps are allowed to only access the contents of their own `/proc/PID` private directory (<https://developer.android.com/about/versions/nougat/android-7.0-changes.html>), so this method does not work anymore for AOSP. The only other alternative for an app to have this kind of access is to request the system-level permission `PACKAGE_USAGE_STATS` (<https://developer.android.com/reference/android/app/usage/UsageStatsManager.html>). The fact that their attack does not apply for passing messages in Android is also proved by the authors' statement that in Android they managed just to launch a resource depletion attack against the browsers. Moreover, the aforementioned restrictions in Nougat prevent apps from accessing `/proc/net` which could otherwise reveal the domain names but not the full URL a user has visited.

Notably, developers in many occasions, despite Google's recommendations (<https://developer.android.com/training/articles/user-data-ids.html>), use `ANDROID_ID` as a unique identifier. To restrict this, Google required that apps request the dangerous permission `READ_PHONE_STATE` (https://developer.android.com/reference/android/Manifest.permission.READ_PHONE_STATE). Clearly, since this ID is unique, installed apps may identify instances and correlate users and behaviours. Since such actions violate user privacy, even though they are performed locally only among installed apps, in the latest preview of Android O, Google decided to block this behaviour so that each app receives a different `ANDROID_ID`. More precisely, in Android O for each combination of application package name, signature, user, and device, developers end up with a different `ANDROID_ID` (<https://developer.android.com/preview/behavior-changes.html>). To further support users controlling their unique identifiers, Google has recently announced the new changes coming in Android O [20], regarding device identifiers. In this regard, Android O is limiting the use of device-scoped identifiers that are not resettable and is also updating the way that applications request account information, providing more user-facing control. The latter signifies that Google is not only aware of such deanonymization issues, but also constantly working on refining its platform to mitigate these threats and restrict unauthorized and unregulated app-to-app communication, let alone web-to-app communication.

Finally, as reported in [21], there are alternative approaches to `ANDROID_ID`. These methods include, but are not limited to, application metadata in the installation folders or metadata from the `procfs` file system. Nevertheless, all these IDs are related to apps and cannot be used to create an ID that a web page could normally have access to.

2.2. Ad Networks. The freemium model is currently the default monetization method in both web services and native Android apps. The main concept of this model is that users may obtain a product which comes in the form of a service or an app for free in some exchange from the user, which is not directly monetary. In the initial form, the trade involved the user having to watch specific ads; however, in the current form, the model monetizes the data which are generated by the user by using the app or service or the ones that are collected from the user, directly or indirectly. This approach has led many to question the ethicality of this model as the actual product is the user and not the app or service.

To clarify the issue one needs to understand that by correlating a considerable amount of information about a user a lot of sensitive information, hence valuable, can be extracted. For instance, by usage statistics one can determine the interests and preferences of a user, when the user may need or want a specific product or service and therefore create a very fine-grained profile for him that is generated without his consent nor his knowledge. In turn, the companies that can collect these data may sell them for, e.g., targeted advertising, tailored to the exact profile of their users, drastically increasing their success.

The above have radically changed the app and web industry, making ad networks among the most highly valued and influential sectors in these fields. In terms of Android apps, the most widely used ad library is Google's Admob; nonetheless, apps often use more than one. In many occasions, ad libraries have proven not to be benign and to exploit the permissions that they have. Note that due to inheritance the ad libraries have the same permissions that are granted to the apps. Furthermore, it should be highlighted that since ad networks are the sole monetization method for freemium apps, developers are following the wills and commands of ad networks by constantly requesting more and more permissions from their users to collect even more data from them.

Stevens et al. [22] found that some of them would use undocumented permissions, read/write to calendar or access location and camera. Grace et al. [23] found that about half of them would probe the corresponding apps to determine whether they could abuse them to harvest sensitive user information. Ads may perform WiFi scans to determine users' location, scan whether the user has accounts in social networks, or even scan the device to find which applications have been installed [24]. In a more sinister scenario, ad libraries try to link devices by playing inaudible sounds [25]. All the above have led researchers to introduce methods to detain them and restrict the access of ad networks and their user profiling methods [26–29]. Notwithstanding their invasiveness, to the best of our knowledge, none of them has been able to pass information from a browser to an app within the Android system. Instead, this kind of communication, whenever reported, was strictly among apps that used the same ad network.

2.3. Web Fingerprinting Techniques. One of the initial ways to track users was through browser cookies. While they can

easily be removed, it was shown that they could be recovered using the so-called *respawning* method, either using Adobe Flash [30], or using ETags and the HTML5 localStorage API [31]. Moreover, a passive network observer could use third-party HTTP tracking cookies to identify users [32].

More advanced methods try to exploit specific characteristics of either the browser or the device to determine whether a specific browser or device has visited a web page in the past. Typical examples involve browser profiling through collecting information like user agent, installed plugins, supported fonts, time zone, language, etc. Depending on the build and user configuration, these characteristics can be used to identify a browser. Nonetheless, many of these characteristics may change due to updates or user intervention.

A more sophisticated approach involves *canvas fingerprinting*, introduced by Mowery and Shacham [33]. The basic concept is that depending on the underlying operating system, font library, graphics card, graphics driver, and browser, a text or an image can be rendered differently. An adversary could track these changes to derive a browser fingerprint. In fact, such characteristics could be correlated across browsers of the same device as the many WebGL characteristics that can be extracted from each browser offer enough entropy for anonymization [34].

However, such variations can also be traced via other methods. More precisely, regarding smartphones it has been shown that hardware components such as accelerometers, speakers, and microphones may have unique characteristics which differ not only from model to model, but also across devices [35–37]. Obviously, since these characteristics are hardware based, they cannot change and are therefore more robust than typical browser specific methods, realizing the concept of *device fingerprinting*.

3. Session Fingerprinting

When someone browses the Internet, his session is considered anonymous unless he has authenticated himself by logging-in, in a web page. While this anonymity is very convenient for ensuring users' privacy, companies strive to find ways to bypass it and to profile them. Frequently, the benign goal behind these actions is usually regarded as the adaptation and/or personalization of a web page according to the corresponding user profile, which translates to better usability and increased content quality, which in turn may increase both views and viewers. In most of the cases, this personalization also targets the advertising industry, since by deanonymizing an individual a business is able to display ads tailored to users' preferences and therefore to increase both business' and service providers' profits.

One of the most widely used methods in achieving this is browser fingerprinting, which tries to deanonymize users by exploiting noticeable differences in the usage of different browsers such as the underlying OS, user agent, browser version, monitor size, or even installed fonts and plugins [33, 38, 39]. More advanced methods go a step further by exploiting device specific variations to identify individual devices. For smartphones, it has been shown that sensors,

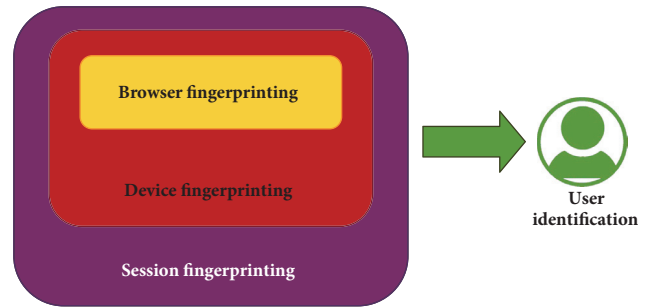


FIGURE 3: Session fingerprinting compared to other methods in the literature.

such as accelerometers, or speakers and microphones may have unique characteristics which differ, not only across models, but also across the devices within which they operate due to calibration errors and frequency distortions [35–37].

While these methods have been proven efficient in many cases, they are usually subject to errors and software updates which could render a previous fingerprint useless. For instance, a browser update may change the user agent or the fonts, making its linking to the previous fingerprint impossible. However, what a company actually needs is to be able to correlate information with other affiliated parties in order to determine whether the user has been simultaneously operating another *session*. A typical example can be regarded as the parallel usage of a web page and a mobile app. Note that while the latter implies that the app is running in the background, it is a typical situation in almost all mobile OSes. In this regard, both parties should try to create a unique ID for each session and also communicate it with each other in order to deanonymize the user. We name the methods for extracting these IDs as *session fingerprinting*. Figure 3 illustrates the relation of our methods to others in the literature. In principle, these methods identify a device; therefore they include device fingerprinting methods; however, since they also include unique identifiers that are collected at each session that a user has with a web page, they amplify the former.

Apparently, these methods depend on the existence of cooperating apps in the mobile device. We argue that this is a weak assumption as the considered adversary in this work is mainly an ad network. Due to the prevalence of the freemium model in Android, most applications are free and, most of the times, they come with at least one preinstalled ad component. However, our requirements do not imply escalated privileges; hence the resulting applications are easier to be accepted by the users.

Although both browsers and the Android OS have such privileges, namely, are able to deanonymize the users and have data about them coming from multiple sources, the level of trust a user has to both of them is the highest possible. Users have chosen them because they trust that they will act honestly and they will protect them from threats and, above all, they will not stalk them. Moreover, it is important to highlight that despite the OSes restrictions, the different existing ad frameworks may indeed perform user profiling,

TABLE 1: Capabilities of apps and web pages.

| | Native Apps | Instant Apps | Web Pages |
|--|-------------|--------------|-----------|
| Access Device Identifiers | ✓ | | |
| Device External Storage | ✓ | | |
| Push Notifications | ✓ | | |
| List of Installed Apps | ✓ | | |
| Access Body Sensors | ✓ | | |
| Direct Communication with Installed Apps | ✓ | | |
| Receiving Broadcasts from OS or 3rd party Apps | ✓ | | |
| Run on the Background | ✓ | | |
| Change Device Settings | ✓ | | |
| Access User Calendar | ✓ | ✓ | |
| Access motion sensors | ✓ | ✓ | |
| Access Contacts | ✓ | ✓ | |
| Access Phone Calls | ✓ | ✓ | |
| Access Sensors | ✓ | ✓ | |
| Access environmental sensors | ✓ | ✓ | |
| Access Location | ✓ | ✓ | ✓ |
| Access Microphone | ✓ | ✓ | ✓ |
| Access position sensors | ✓ | ✓ | ✓ |
| Access Camera | ✓ | ✓ | ✓ |
| Access Internal Storage (own storage) | ✓ | ✓ | ✓ |
| High precision timestamps | ✓ | ✓ | ✓ |

but still they cannot escape the browser or the app ecosystem within which they operate.

4. Problem Setting

The introduction of WWW is one of the milestones in modern computing, enabling users all over the globe to collect and share information with their peers. In fact, the emergence of Social Networks and Media has further reshaped the landscape. In principle, the information that can be collected from a web page is derived solely via the launched browser and is strictly limited to the browser's environment. Any attempt to access resources beyond the browser sandbox is considered as a security violation and therefore is characterized as malicious. To this end, browsers allow very limited exposure of user data to a web page. For instance, a web page cannot read from or write to the storage of a mobile device unless this action is user initiated. To overcome these restrictions, adversaries may resort to browser extensions [40] which provide even more capabilities. On top of that, nowadays, due to the growth of mobile devices, several standards, like HTML5, have been introduced to allow browsers to access additional resources, such as location, camera, or microphone, upon direct and explicit user consent. As a result, web pages have a growing set of capabilities, yet quite limited in comparison to apps. It is worth noting, however, that Chrome, the default Android Browser, as well as many other browsers do not support extensions in the mobile environment, even though they have numerous desktop editions. Finally, up to recently, Chrome apps, also available only for desktop installations,

are discontinued as of early 2018. Therefore, browsers in mobile environments have significantly less functionality and extendability than their desktop peers.

On the other hand, Android apps reside on a different environment. Contrary to web pages, mobile apps “live” directly in the operating system and thus have more direct access to its hardware and corresponding resources. Again, their access is limited according to the granted privileges by the users plus their scope is more fixed as they fulfill specific user needs. Due to this restriction, apps cannot determine which web pages a user visits. A critical distinction between Android apps and web pages is that apps always pass through an “installation” process. This step represents a user acknowledgment regarding the specific resources an app is allowed to use inside the environment executed. Notably, Android Marshmallow users may grant and revoke permissions to specific resources, like camera, microphone, location, etc., which are called “dangerous” and may hinder security and privacy issues. On the contrary, this is not the case for web pages where users are not faced with preconditions for visiting them. In many instances, users would like to be able to use “one-time apps” to accomplish specific tasks like using a retailer’s app when browsing his web page. To address this need, Google recently introduced *instant apps*, which do not require installation and have more permissions and/or capabilities than common web pages. However, instant apps, like web pages, are also restricted from accessing hardware identifiers to prevent user profiling.

Key differences in terms of the capabilities of Android apps, native and instant, and web pages are illustrated in Table 1. As expected from the earlier discussion, it can

be easily noticed that installed applications have far more access to device resources than web pages, since users install apps granting themselves the corresponding permissions. On the contrary, due to the nature of the web, users may visit a large number of different web pages on a daily basis, without knowing their quality, intentions, source, or content. Hence, both Android and browsers make significant efforts, e.g., running in a sandbox environment, towards protecting users from malicious web page behaviour. Unquestionably, if an app had been able to communicate with a web page without restrictions, the entire underlying security infrastructure would have been rendered useless. In fact, even the most widely used apps in Android are not able to communicate directly with their web page. For instance, in the case of three well-known and widely used apps, Facebook, Instagram, and Twitter, they do not transfer information to their corresponding web page or any other cooperating web page when a user has logged in the app. Instead, a “Connect with Facebook/Twitter” button usually appears, requiring further user interaction and most importantly being realized by users. Evidently, had these apps been able to transfer this kind of information to the browser, they would have done it already long time ago, not only for facilitating users, but also for further increasing the amount of collected user data and the quality of provided services.

Creating a mechanism being able to transmit an identifier from the browser to a cooperating installed app in a user’s device, or vice versa, would allow for the installed app to identify the individual who visited the cooperating web page and subsequently the web page would elevate its access to the same resources as those of the installed app. Further analyzing this, after both parties, namely, web pages and apps, have identified themselves lying in the same user’s device, they would be able to create a covert channel. In the least sinister scenario, a web page cooperating with an app would be able to access a user’s contacts, SMS messages, or even storage and microphone, without obtaining user’s consent, and would manage to display ads perfectly tailored to the user’s profile, albeit violating his privacy. However, in a true malicious scenario, user data would be harvested by web pages and personalized exploits would be pushed to users’ devices to further exploit their personal data while they surf in the WWW.

5. Intercommunication between Apps and Web Pages

In the following subsections, we provide a set of concrete examples as Proofs of Concept, which showcase how apps and web pages can mutually create and consequently transmit unique IDs that allow them to link their usage and to communicate sensitive attributes to each other, realizing what the authors of this paper have introduced as “session fingerprinting”. The authors of this paper have responsibly disclosed the mentioned security issues described in the next subsections, regarding unauthorized communications between apps and web pages.

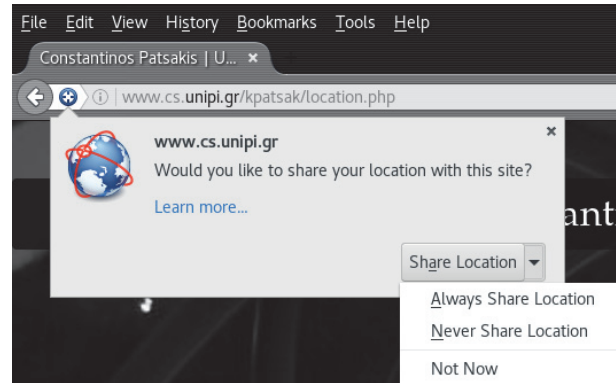


FIGURE 4: Firefox requests user’s permission to allow a web page to access location.

5.1. Location. Location awareness has undoubtedly increased the potential of many applications since it allows them to adapt accordingly and render their information based on location specific criteria, drastically improving, e.g., user recommendations. Obviously, location is a sensitive piece of information as it can disclose many private attributes, ranging from work and residence location, to entertainment preferences and political/religious beliefs if correlated with other sources of information. Therefore, mobile OSes allow applications to access location data only if the user grants a corresponding permission. In Android this permission is provided either as *fine* or as *coarse* location.

Similarly, beyond the support for media in HTML5, the standard enables web pages to access user location. Since this information is sensitive, the browser specifically requests user permission to be granted—see Figure 4—even though this kind of information can be used for other purposes as well. Once the browser gets access to the user’s location, the response contains apart from the longitude and the latitude the accuracy and the timestamp [41] as well. Moreover, depending on the implementation, it may also return other values, such as heading and speed. Interestingly, in our research we have come up with proofs that this information can be correlated with location data information from an Android app. More precisely, while an Android app could monitor a user’s location and is able to correlate the coordinates with the ones that are received from a web page, one could argue that since these requests to location data are not made simultaneously, from the browser and the app, the actual identity of the user is not disclosed. This argument can be clearly supported either because other nearby users may be also implicated, or because the web page gets this information only once. However, practically this is not the case. For reasons such as minimizing battery consumption, since the usage of the GPS is rather greedy, Android app developers may choose to use the “last known location” feature through the `getLastLocation()` method of `LocationServices` which fetches the location from its cache [42]. Based then on the accompanied timestamp the developer can determine whether he needs to request a new reading or not. Yet, what seems quite interesting in this case is that our findings reveal

that, by accessing the device's last known coarse location from an app, we actually end up having data about the precise last location request made by a web page. It should be emphasized that "coarse" location is the one that should be used here, since "fine" location is accessed exclusively by Android apps and not web pages and hence is not suitable for our method.

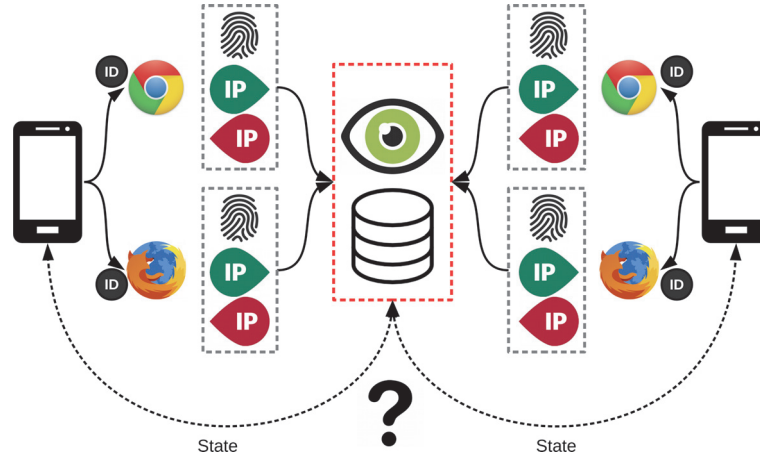
Apparently, if a mobile app monitors the last known location and its corresponding timestamp determined by the Network Location Provider and communicates this information to the "All Seeing Eye", the latter is able to determine whether it coincides with user's coordinates and timestamp received from a web page. Beyond a doubt, this combination of data is quite "unique". Once a correlation is found, the All Seeing Eye can create a covert channel that will serve both the app and the web page to exchange data regarding this session. This specific instance of "session fingerprinting" has been successfully tested in all recent Android versions, from Marshmallow to Oreo. Nevertheless, it is expected to be fully functional to all Android versions, since it utilizes one of the most basic and native Android mechanisms, namely, location services, available since API level 1. Geolocation W3C API of HTML5 has been also been available since the first versions of mobile browsers, namely, Android Browser, Samsung Internet, and Google Chrome [43].

5.2. Browser Fingerprinting. In the previous example a dangerous, yet very commonly used permission, namely, location, was used to identify a user. Nevertheless, one could achieve the same result without such permissions. A more stealth method is to utilize browser fingerprinting. To this end, we assume that the victim has installed an application which does not request any dangerous permission. According to the Android permission model, such applications are allowed to list all the installed applications in a device; hence the adversary has also knowledge of all the installed browser applications. This way, the app can subsequently open all the available browsers through intents and point them to a desired URL in order to obtain a fingerprint from them. Note that, as for Nougat, an application cannot determine which is the foreground application, a piece of information that would have been very valuable for the adversary; however, the authors of this paper have already notified Google of a new method to achieve this in all versions prior to Nougat (Android Issue no 23504, triaged). Each time a browser is fingerprinted by the app, a random nonce is created and is sent in the web page request allowing the adversary to determine to whom its fingerprint belongs. This kind of attack utilizes malicious intents, while for "covering traces" purposes the cooperating malicious web pages could be redirected to a commonly used web page (e.g., a search engine), after accomplishing the ID exchanging job. A scenario where no intents are needed also exists, where a malicious app may use its native webview component in order to accomplish the aforementioned task.

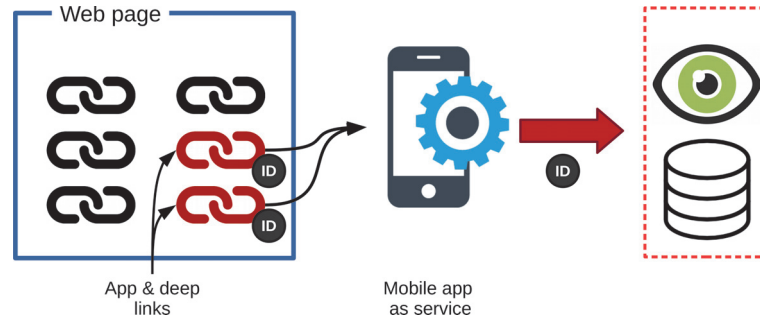
Since mobile devices have less "unique" characteristics compared to personal computers, an extension to browser fingerprints is to additionally use even more mobile device

characteristics, further conforming to the "session fingerprinting" proposed term. Indeed, both mobile app and web page can obtain knowledge about the internal and the external IP of the mobile device. For the former one could potentially use WebRTC [44] which is known to leak several pieces of private information [45]. Therefore, when a user visits a web page, the web page queries the "All Seeing Eye" to determine if someone with the specific browser fingerprint and public and local IPs has a cooperating app running at a specific timeframe. In general, the chances of this query returning more than one result are slim. Nonetheless, in a corporate environment, where many people might have mobile devices of the same model, some instances may exist. In such environments one could have two identical devices with the same internal IP, if, e.g., two users with the same smartphone model use the corporate WiFi on different floors or departments. To further reduce the query results, the "All Seeing Eye" could request the state of each device. In this case, additional information that can be cross-checked between apps and web pages includes, but is not limited to, the following: battery information (both state and charging level), interval since device's last noticeable movement (this could be determined, e.g., via accelerometers), interval since last proximity (via proximity sensor), light measurements, positioning (e.g., facing up or down), or even some connection statistics such as `downlinkMax` (one of the new features of HTML5 through the Network Information API [46]). From this information one can easily determine which user is using a smartphone at a specific timeframe and essentially eliminate the possibilities of having false positives. This process is illustrated in Figure 5(a). In this figure, we may notice that both web pages inside browsers and also web pages inside applications' webview components are able to collect unique fingerprints, namely, internal and external IPs, accompanied by information regarding the devices' state and communicate them to the "All Seeing Eye" which will then be responsible for finding exact matches between them. Due to the huge amount of data that have to be correlated, approaches like [47] can be used to boost the performance. HTML5s WebRTC W3C API is available to all Android smartphones running Samsung Internet browser and Google Chrome. Network Information API is available on Android smartphones with Android Internet browser version ≥ 2.2 and Google Chrome version ≥ 38 , [43].

5.3. App and Deep Links. Most users install plenty of apps on their devices, even though many of them might have some overlapping functionality. To facilitate user interaction between websites and Android applications the Web Intents framework was introduced, allowing a developer to specify how a hyperlink is handled on the user device, e.g., open the phone to dial up an already prepared number or use Skype for a specific contact. However, Android supports further features through *app* and *deep links*. The concept behind both of these types of web links is to open specific apps depending on the link. As an example, Facebook and Twitter apps are triggered when the user taps on a link referring to content of the corresponding site.



(a) Identification through browser fingerprinting



(b) Identification through app and deep links

FIGURE 5: User identification methods.

Interestingly, this kind of functionality is automatically activated when a user installs an application which provides such features, without requesting any user approval. Moreover, Android activities may run on the foreground in a “hidden” mode, either by using transparent themes or by utilizing floating zero-sized activities. Practically, this creates a hidden communication channel between web pages and apps that can be used to identify users as illustrated in Figure 5(b). We assume that an app is installed in a user’s device having at least one “browsable” (declared inside Apps Manifest file) activity in order to enable “cooperation” with web pages. On the other side, web pages embed some special “intent” hyperlinks which also have the ability to carry a random ID, different for every interaction. Once a user taps in one of these links, the ID is bundled and transferred to the app, using the “getExtras” method inside the “Intent” Android class. As a final step, once again the data is communicated to the All Seeing Eye, deanonymizing the user. This kind of “interaction” can be seen as a directed web-to-app communication, where an app has the ability to be reached from web pages. At the same time one or more web pages may utilize this “functionality” by transmitting an ID which is essential for the entire described scenario. Next subsection describes how this local channel communication can be achieved through the opposite direction of interaction. Regarding “browsable” activities that enable deep linking, this

feature is available in Android since API level 1. As a result, this instance of “session fingerprinting” is also expected to affect all versions of Android.

5.4. Direct App to Web Communication. Following the same logic as in the previous subsections, an app is also able to directly reach a web page through a device’s installed browser. Once again, Android Intents are deployed in order to launch an installed browser and to pass a specific URL. The cooperating app is able to inject one or more string values inside the URL as parameters, which in our case is a simple, randomly generated ID, and correspondingly fire a malicious intent towards a browser. As a next step, the loaded web page can extract the ID from the URL’s “location search” property and thus a local covert channel between the app and the launched web page is realized. Naturally, the web page is again able to communicate the ID to the “All Seeing Eye”, making the user’s profile available to others as well. As already discussed, this kind of method “leaves some traces”; namely, it opens a web browser; however the corresponding malicious web page can hide its traces with a simple redirection, e.g., to a search engine’s default page.

A quite significant detail in this process is that the corresponding web page should initially use a client side programming language to retrieve the transmitted ID. This is essential in order to create the local covert channel between

TABLE 2: Privacy exposure to the described threats per API level.

| Method | API level |
|---------------------------------|---------------------------------------|
| Location | > 1 |
| Browser fingerprinting | Browser specific. Native WebView > 20 |
| App and deep links | > 1 |
| Direct App to Web communication | > 1 |

TABLE 3: Necessary permissions for each Android ad network.

| | % of installing | Internet | ACCESS_NETWORK_STATE | WRITE_EXTERNAL_STORAGE | ACCESS_WIFI_STATE |
|------------|-----------------|----------|----------------------|------------------------|-------------------|
| Admob | 61.73 | ✓ | | | |
| Unity Ads | 19.02 | ✓ | ✓ | | |
| Chartboost | 14.07 | ✓ | ✓ | | |
| MoPub | 13.62 | ✓ | ✓ | | |
| AdColony | 13.18 | ✓ | ✓ | | |
| AppLovin | 12.91 | ✓ | | ✓ | |
| AppsFlyer | 10.23 | ✓ | ✓ | ✓ | ✓ |
| InMobi | 9.14 | ✓ | ✓ | | |
| Vungle | 7.08 | ✓ | ✓ | ✓ | |
| Tapjoy | 6.72 | ✓ | ✓ | | ✓ |

TABLE 4: Results from Tacyt.

| | Google Play | | Other markets | |
|--------------------------|-------------|-------------|---------------|-------------|
| | Available | Unavailable | Available | Unavailable |
| App & Deep links | 432,204 | 87,483 | 71,686 | 34 |
| ACCESS_COARSE_LOCATION | 996,326 | 215,335 | 154,749 | 29 |
| INTERNET | 4,044,922 | 1,046,310 | 533,492 | 149 |
| Total versions in market | 4,207,542 | 1,095,398 | 576,204 | 155 |

the app and the web page, since an app ID directly transmitted to a web server would lose the ability to “find its way back” to the corresponding device’s web page. This fourth instance of “session fingerprinting” also utilizes native Android mechanisms, since even the first Android smartphones were able to use intents and communicate with web pages through browsers. As a result, it has been successfully tested on all recent Android versions, from Marshmallow to Oreo.

6. Experimental Results and Statistics

In Table 2 we illustrate from which API levels Android devices are exposed to the threats we have described, indicating that these methods apply to the vast majority of devices available.

In order to provide an estimation of the potential exposure of users to these threats, the authors of this paper used data by utilizing “Tacyt” (<https://tacyt.elevenpaths.com>). Tacyt is an innovative cyber intelligence tool that facilitates research in Android mobile apps environments with big data technology. The aim of Tacyt is to enable quick detection, discovery, and analysis of these threats in order to reduce their potential impact on organizations. Enabling app data mining and detection enables research and analysis of the collected information from Google Play and other markets. Due to the

implementation of Tacyt, the query responses are per app version and not per app; nonetheless they provide a very good overview of apps dating back to at least three years ago.

Table 4 presents the results from the performed queries. In our first query we tried to identify how many apps provide a deep or app link. This information is declared in the manifest of each application and is clearly marked with the `android.intent.category.BROWSABLE` tag of the XML file. The next two rows involve app versions which required the `ACCESS_COARSE_LOCATION` and Internet permission which could be potentially used to deanonymize users. Similarly, using PublicWWW, a source code search engine for web pages, we found more than 96,000 web pages to use geolocation features in their code for locating users. A sample of the attributes received from both APIs is illustrated in Box 1.

Regarding ad networks in Android, we tried to highlight the requirements, in terms of permissions, that each of them request from the developers and whether they could exploit our methods. Analyzing the requested permissions of the top 10 ad networks in Android according to Appbrain (<https://www.appbrain.com/stats/libraries/ad>) (see Table 3 for the corresponding list), we found that all of them required the obvious normal permission of `Internet`. Moreover,

```

float getAccuracy(): Get the estimated horizontal accuracy of this location, radial, in meters.
double getAltitude(): Get the altitude if available, in meters above the WGS 84 reference ellipsoid.
float getBearing(): Get the bearing, in degrees.
float getBearingAccuracyDegrees(): Get the estimated bearing accuracy of this location, in degrees.
long getElapsedRealtimeNanos(): Return the time of this fix, in elapsed real-time since system boot.
Bundle getExtras(): Returns additional provider-specific information about the location fix as a Bundle.
double getLatitude(): Get the latitude, in degrees.
double getLongitude(): Get the longitude, in degrees.
String getProvider(): Returns the name of the provider that generated this fix.
float getSpeed(): Get the speed if it is available, in meters/second over ground.
float getSpeedAccuracyMetersPerSecond(): Get the estimated speed accuracy of this location, in meters per second.
long getTime(): Return the UTC time of this fix, in milliseconds since January 1, 1970.
float getVerticalAccuracyMeters(): Get the estimated vertical accuracy of this location, in meters.
boolean hasAccuracy(): True if this location has a horizontal accuracy.
boolean hasAltitude(): True if this location has an altitude.
boolean hasBearing(): True if this location has a bearing.
boolean hasBearingAccuracy(): True if this location has a bearing accuracy.
boolean hasSpeed(): True if this location has a speed.
boolean hasSpeedAccuracy(): True if this location has a speed accuracy.
boolean hasVerticalAccuracy(): True if this location has a vertical accuracy.
boolean isFromMockProvider(): Returns true if the Location came from a mock provider.

Position.coords: Returns a Coordinates object defining the current location.
Position.timestamp: Returns a DOMTimeStamp representing the time at which the location was retrieved.
Where coordinates contain the following information:
double Coordinates.latitude: The position's latitude in decimal degrees.
double Coordinates.longitude: The position's longitude in decimal degrees.
double Coordinates.altitude: The position's altitude in meters, relative to sea level. Can be null
if the implementation cannot provide the data.
double Coordinates.accuracy: he accuracy of the latitude and longitude properties, expressed in meters.
double Coordinates.altitudeAccuracy: he accuracy of the altitude expressed in meters. This value can be null.
double Coordinates.heading: he direction in which the device is traveling in degrees.
double Coordinates.speed: he velocity of the device in meters per second. This value can be null.

```

Box 1: Sample attributes from native Android and JavaScript geolocation APIs.

most of them (8/10) requested the `ACCESS_NETWORK_STATE` permission. Notably, 3 of them requested access to dangerous permissions, while all of them requested writing data in the device (`WRITE_EXTERNAL_STORAGE`). Finally, all of them also proposed the use of location (`ACCESS_COARSE_LOCATION/FINE_COARSE_LOCATION`).

Combining the evidence from the sections regarding session fingerprinting and this section's experimental data regarding ad network permissions, one may easily infer that our proposed techniques for user deanonymization are realistic in terms of permission requirements, since they require either zero or normal permissions or, in some cases, dangerous permissions that are frequently requested by ad networks. What is more alarming is the concern whether this paper sayings are already applied and utilized by existing ad networks, third-party apps, and corresponding web pages. Interestingly, after analyzing the above information we may discuss that imposing restrictions or even applying control mechanisms for the Internet could result in a large benefit towards the users' privacy. Nevertheless, this special, "normal", permission seems more tightly linked to all kinds of advertising and marking it as "dangerous" would require more than strong will by the companies involved.

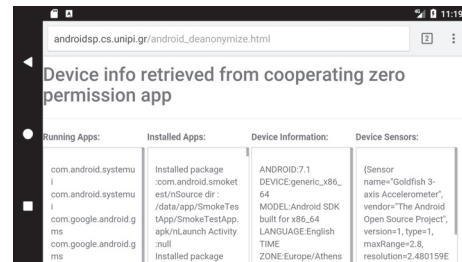


FIGURE 6: Device info as obtained from a web page through a zero-permission app installed in a device running Android 7.1.1.

Finally, we have implemented a proof of concept app that is able to cooperate with web pages without requesting any dangerous permissions. The app is available at androidsp.cs.unipi.gr/android_deanonymize.html. Once installed, the app recovers a lot of information from the device, such as installed and running apps and device info as well as measurements from many sensors which do not require any dangerous permissions. Eventually, as illustrated in Figure 6, when the user visits the corresponding web

page through his mobile browser he can verify that the web page has recovered all this information without requesting his consent. It is worth noting that, apart from providing access to sensors that a web page would not normally have, the measurements for these sensors are also listed in a fine-grained mode, e.g., access to accelerometer detailed measurements which could be used for fingerprinting [35].

7. Conclusions

The ever increasing use of mobile devices exposes user privacy in numerous ways. Despite the fact that mobile OSes take several measures to protect their users, attackers seem to always be one step ahead. Nonetheless, most would agree that the state-of-the-art countermeasures guarantee an independence between the browser and the mobile apps so they cannot exchange information. Taking Android as our reference platform, we introduce new methods that exploit various inherent mechanisms to practically guarantee absolute identification with limited resource usage. Moreover, the proposed methods extend the notion of device fingerprinting to what we have introduced as “session fingerprinting”. Our techniques can be performed without accessing unique device characteristics or using dangerous permissions. In this regard, our techniques imply a bigger threat, as the covert channel that is created between the web pages and the apps cannot be easily traced.

Due to the fact that all the aforementioned mechanisms are inherent in Android, one cannot rule out the possibility that these mechanisms are already being exploited, enabling unauthorized and unregulated cooperation between the two ecosystems. Clearly, this would greatly expose users’ privacy, bypassing the permission model of the most widely used mobile platform to date. Addressing such issues is a rather challenging task, because, apart from changing the native Android mechanisms, it is also a requirement for the OS to determine the context of some function calls, either to prohibit access to resources or to obfuscate the underlying information, since these calls might seem legitimate.

Conflicts of Interest

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

Acknowledgments

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

References

- [1] The Economist, “The world’s most valuable resource is no longer oil, but data,” 2017, <https://www.economist.com/news/leaders/21721656-data-economy-demands-new-approach-anti-trust-rules-worlds-most-valuable-resource/>.
- [2] E. Alepis and C. Patsakis, “The All Seeing Eye: Web to App Intercommunication for Session Fingerprinting in Android,” in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, vol. 10656 of *Lecture Notes in Computer Science*, pp. 93–107, Springer International Publishing, 2017.
- [3] Or. Peles and Roe. Hay, “One class to rule them all: 0-day deserialization vulnerabilities in android,” in *Proceedings of the 9th USENIX Workshop on Offensive Technologies (WOOT 15)*, USENIX Association, 2015.
- [4] E. Alepis and C. Patsakis, “Trapped by the UI: The Android Case,” in *Research in Attacks, Intrusions, and Defenses*, vol. 10453 of *Lecture Notes in Computer Science*, pp. 334–354, Springer International Publishing, 2017.
- [5] E. Alepis and C. Patsakis, “Hey Doc, Is This Normal?: Exploring Android Permissions in the Post Marshmallow Era,” in *Security, Privacy, and Applied Cryptography Engineering*, vol. 10662 of *Lecture Notes in Computer Science*, pp. 53–73, Springer International Publishing, 2017.
- [6] E. Alepis and C. Patsakis, “There’s Wally! Location Tracking in Android without Permissions,” in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, pp. 278–284, Porto, Portugal, February 2017.
- [7] E. Alepis and C. Patsakis, “Monkey Says, Monkey Does: Security and Privacy on Voice Assistants,” *IEEE Access*, vol. 5, pp. 17841–17851, 2017.
- [8] Q. Do, B. Martini, and K.-K. R. Choo, “Exfiltrating data from Android devices,” *Computers & Security*, vol. 48, pp. 74–91, 2015.
- [9] P. McDaniel, “Bloatware comes to the smartphone,” *IEEE Security & Privacy*, vol. 10, no. 4, pp. 85–87, 2012.
- [10] H. Elahi, G. Wang, and L. Xu, “Smartphone bloatware: An overlooked privacy problem,” in *Proceeding of the Security, Privacy, and Anonymity in Computation, Communication, and Storage - 10th International Conference, (SpaCCS ’17)*, G. Wang, M. Atiquzzaman, Z. Yan, and K. K. R. Choo, Eds., vol. 10656 of *Lecture Notes in Computer Science*, pp. 169–185, Guangzhou, China, 2017.
- [11] L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, “Privilege escalation attacks on android,” in *Information Security*, vol. 6531, pp. 346–360, Springer, 2011.
- [12] C. Orthacker, P. Teufl, S. Kraxberger et al., “Android security permissions—can we trust them?” in *Security and Privacy in Mobile Information and Communication Systems*, pp. 40–51, Springer, 2012.
- [13] A. Dimitriadis, P. S. Efraimidis, and V. Katos, “Malevolent app pairs: an android permission overpassing scheme,” in *Proceedings of the ACM International Conference on Computing Frontiers, CF ’2016*, pp. 431–436, ACM, New York, NY, USA, May 2016.
- [14] F. Vincent Taylor, R. Alastair Beresford, and Ivan. Martinovic, “Intra-library collusion: A potential privacy nightmare on smartphones,” *arXiv preprint arXiv:1708.03520*, 2017.
- [15] J. Blasco and T. M. Chen, “Automated generation of colluding apps for experimental research,” *Journal of Computer Virology and Hacking Techniques*, pp. 1–12, 2017.
- [16] A. Feizollah, N. B. Anuar, R. Salleh, G. Suarez-Tangil, and S. Furnell, “AndroDialysis: Analysis of Android Intent Effectiveness in Malware Detection,” *Computers & Security*, vol. 65, pp. 121–134, 2017.
- [17] W. Gasior and L. Yang, “Exploring covert channel in android platform,” in *Proceedings of the ASE International Conference on Cyber Security (CyberSecurity ’12)*, pp. 173–177, Washington, DC, USA, December 2012.

- [18] S. Chandra, Z. Lin, A. Kundu, and L. Khan, "Towards a Systematic Study of the Covert Channel Attacks in Smartphones," in *International Conference on Security and Privacy in Communication Networks*, vol. 152 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 427–435, Springer International Publishing, 2015.
- [19] M. Rushanan, D. Russell, and A. D. Rubin, "MalloryWorker: Stealthy Computation and Covert Channels Using Web Workers," in *Security and Trust Management*, vol. 9871 of *Lecture Notes in Computer Science*, pp. 196–211, Springer International Publishing, 2016.
- [20] Android Developers Blog, "Changes to device identifiers in android o," 2017, <https://android-developers.googleblog.com/2017/04/changes-to-device-identifiers-in.html>.
- [21] E. Alepis and C. Patsakis, "Persistent vs Service IDs in Android: Session Fingerprinting from Apps" in *Mobile Networks and Management*, vol. 235 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 14–29, Springer International Publishing, 2018.
- [22] R. Stevens, C. Gibler, J. Crussell, J. Erickson, and H. Chen, "Investigating user privacy in android ad libraries," in *Proceedings of the 2012 Workshop on Mobile Security Technologies*, 2012.
- [23] M. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi, "Unsafe exposure analysis of mobile in-app advertisements," in *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec'12*, pp. 101–112, USA, April 2012.
- [24] T. Book, A. Pridgen, and D. S. Wallach, "Longitudinal analysis of android ad library permissions," *arXiv preprint arXiv:1303.0857*, 2013, <https://arxiv.org/abs/1803.03270>.
- [25] D. Goodin, "Beware of ads that use inaudible sound to link your phone, tv, tablet, and pc," 2015w <http://arstechnica.com/tech-policy/2015/11/beware-of-ads-that-use-inaudible-sound-to-link-your-phone-tv-tablet-and-pc/>.
- [26] P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, "AdDroid: privilege separation for applications and advertisers in Android," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS '12)*, pp. 71–72, Seoul, Republic of Korea, May 2012.
- [27] S. Shekhar, M. Dietz, and D. S. Wallach, "Adsplit: Separating smartphone advertising from applications," in *Proceeding of the Presented as part of the 21st USENIX Security Symposium (USENIX Security'12)*, pp. 553–567, 2012.
- [28] X. Zhang, A. Ahlawat, and W. Du, "AFrame: Isolating advertisements from mobile applications in android," in *Proceedings of the 29th Annual Computer Security Applications Conference, (ACSAC '13)*, pp. 9–18, December 2013.
- [29] V. Tsiakos and C. Patsakis, "AndroPatchApp: Taming Rogue Ads in Android," in *Mobile, Secure, and Programmable Networking*, vol. 10026 of *Lecture Notes in Computer Science*, pp. 183–196, Springer International Publishing, 2016.
- [30] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle, "Flash cookies and privacy," in *Proceedings of the 2010 AAAI Spring Symposium*, pp. 158–163, March 2010.
- [31] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle, "Flash Cookies and Privacy II: Now with HTML5 and ETag Respawning," *SSRN Electronic Journal*, 2011.
- [32] S. Englehardt, D. Reisman, C. Eubank et al., "Cookies that give you away: The surveillance implications of web tracking," in *Proceedings of the 24th International Conference on World Wide Web, (WWW '15)*, pp. 289–299, May 2015.
- [33] K. Mowery, S. Keelveedhi, and H. Shacham, "Pixel perfect: Fingerprinting canvas in html5," in *Proceedings of W2SP*, p. 19, Raleigh, North Carolina, USA, October 2012.
- [34] Y. Cao, S. Li, and E. Wijmans, "(cross-)browser fingerprinting via os and hardware level features," in *Proceedings of the 24th Annual Network and Distributed System Security Symposium, NDSS*, San Diego, CA.
- [35] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "AccelPrint: imperfections of accelerometers make smartphones trackable," in *Proceedings of the Network and Distributed System Security Symposium (NDSS '14)*, San Diego, Calif, USA, February 2014.
- [36] Z. Zhou, W. Diao, X. Liu, and K. Zhang, "Acoustic fingerprinting revisited: Generate stable device ID stealthily with inaudible sound," in *Proceedings of the 21st ACM Conference on Computer and Communications Security, (CCS '14)*, pp. 429–440, USA, November 2014.
- [37] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device identification via sensor fingerprinting," *arXiv preprint arXiv:1408.1416*, 2014, <https://arxiv.org/abs/1408.1416?context=cs>.
- [38] P. Eckersley, "How unique is your web browser?" in *International Symposium on Privacy Enhancing Technologies Symposium*, vol. 6205 of *Lecture Notes in Computer Science*, pp. 1–18, Springer, Berlin, Germany, 2010.
- [39] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham, "Fingerprinting information in javascript implementations," in *Proceedings of W2SP*, vol. 2, pp. 180–193, 2011.
- [40] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson, "Hulk: Eliciting malicious behavior in browser extensions," in *USENIX Security*, pp. 641–654, 2014.
- [41] A. Popescu, "Geolocation api specification 2nd edition," 2016, <https://www.w3.org/TR/geolocation-API/>.
- [42] Android developers, "Getting the last known location," 2017, <https://developer.android.com/training/location/retrieve-current.html>.
- [43] Mobile HTML, "Html5 compatibility on mobile and tablet browsers with testing on real devices," 2018, <http://mobilehtml5.org/>.
- [44] A. Bergkvist, D. C. Burnett, C. Jennings, A. Narayanan, and B. Aboba, "WebRTC 1.0: Real-time communication between browsers," 2016, <https://www.w3.org/TR/webrtc/>.
- [45] V. Beltran, E. Bertin, and N. Crespi, "User identity for WebRTC services: A matter of trust," *IEEE Internet Computing*, vol. 18, no. 6, pp. 18–25, 2014.
- [46] M. Cáceres, F. J. Moreno, and I. Grigorik, "Network information API," 2017, <http://wicg.github.io/netinfo/>.
- [47] W. Yang, G. Wang, K. R. Choo, and S. Chen, "HEPart: A balanced hypergraph partitioning algorithm for big data applications," *Future Generation Computer Systems*, vol. 83, pp. 250–268, 2018.

Research Article

Towards a New Algorithm to Optimize IPv6 Neighbor Discovery Security for Small Objects Networks

Ali El Ksimi  and Cherkaoui Leghris

RTM Team, L@M, Faculty of Science and Technologies Mohammedia, University of Hassan II Casablanca, Morocco

Correspondence should be addressed to Ali El Ksimi; ali.elksimi@yahoo.fr

Received 28 December 2017; Revised 20 April 2018; Accepted 30 April 2018; Published 6 June 2018

Academic Editor: Wenjia Li

Copyright © 2018 Ali El Ksimi and Cherkaoui Leghris. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to verify the uniqueness of link-local or unicast addresses, nodes must perform a Duplicate Address Detection process before using them. However, this process is subject to many attacks and the security is willing to be the most important issues in Small Object Networks with IPv6. In this paper, we developed a new algorithm to optimize the security in IPv6-DAD process; this method is based on SHA-512 to verify the identity of the Neighbor Discovery messages transmitted in the link local. First, before sending the NS message, the new node uses the function SHA-512 to hash to the target address and use the last 64 bits in a new field and then encrypt the result with its private key. When receiving the secure message, the existing nodes decrypt it. Our algorithm is going to secure the DAD process by using a digital signature. Overall, this algorithm showed a significant effect in terms of the Address Configuration Success Probability (ACSP).

1. Introduction

The network protocol mainly used today for Internet communications is the Internet Protocol (IP). The IPv4 protocol suffers from many weaknesses such as the insufficient address space nowadays. Indeed, the IPv4 addresses are 32 bits long, which represents about 4,3 milliard of possible IPv4 addresses. Following the explosion of network growth Internet and wastage of addresses due to the class structure, the number of IPv4 addresses has become insufficient. Another problem is the saturation of the routing tables in the main routers of the Internet. Although since 1993, many emergency measures have been taken, this only allows delaying its deadline. So, the Internet Engineering Task Force (IETF) launched work in 1994 to specify the Internet Protocol that will replace IPv4: this protocol is IPv6 [1].

The Neighbor Discovery (NDP) [2] is the most important part in IPv6; it allows a node to integrate into the local network environment in which IPv6 packets are physically transmitted. Through to this protocol, it becomes possible to interact with the equipment connected to the same support (stations and routers). It is important to note that for a given node, the neighbors discovery does not consist in establishing

an exhaustive list of the others connected to the link. Indeed, it is only to manage those with whom it dialogues. This protocol performs the following functions: Address Resolution, Neighbor Unreachability Detection, Autoconfiguration, and Redirect Indication. It uses five messages including Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Announcement, and Indication redirection. The IPv6 Stateless Address AutoConfiguration (SLAAC) [3] of IPv6 is primarily based on the NDP process. This mechanism uses Duplicate Address Detection (DAD) [4] to verify the uniqueness of the addresses on the same link. However, it is vulnerable to attack and many solutions have been standardized to minimize this vulnerability such as SECure Neighbor Discovery (SEND) [5], but they are subject to certain limitations.

We base our study on SmObNet6 (Small Objects Network with IPv6) which is a generic term used to define either small or larger network to connect small communicating objects. The use of IPv6 protocol regarding communication, collecting, and exchanging data between objects represents a common point between these networks within the Internet infrastructure. This paper treats the SLAAC phases and explains the problems associated with them. So, we propose

a new algorithm based on SHA-512 [6] in order to optimize IPv6 security in SmObNet6,

This paper is organized as follows: Section 2 presents a related work to our field when Section 3 describes some IPv6 functionalities, in particular, the DAD process. Section 4 shows the parameters and methodology following in this work and we present our algorithm in Section 4. Section 5 includes the algorithm implementation with evaluation, after we conclude this paper and addresses some prospects.

2. Related Work

Attacks on the IPv6 operations, especially on DAD process, become one of the interesting research fields. Several proposals have been made by researchers to address security issues in IPv6 DAD. Many authors have treated this problem.

In [7], the authors have proposed a scheme to secure IPv6 address which includes the modifications to the RFC 3972 standard by reducing the granularity factor of a sec from 16 to 8 and replacing RSA with ECC and ECDSA, using SHA-256 [6] hash function. This method improves the address configuration performance, but it does not eliminate the address conflict.

In [8], the authors have presented a new algorithm for address generation. This mechanism has a minimal computation cost as compared to CGA. Nevertheless, this mechanism uses SHA-1 hash encryption which is vulnerable to collisions attacks.

In [9], the authors have utilized a novel approach for securing IPv6 link-local communication. They have used an alternative approach for the CGA and SEND protocols which still represent a limitation to the security level.

Another approach such as secure IPv6 address configuration protocol for vehicular networks [10] was proposed to ensure security in IPv6 without DAD process. However, this method is used only when the distance between a vehicle and its serving AP is one-hop.

In [11], the authors have proposed a new method to secure Neighbor Discovery Protocol in IPv6. This mechanism is based on SDN controller to verify the source of NDP packets. However, this method is not efficient because it does not handle the detection of NDP attacks.

Another method was used in [12] to secure the DAD; it is called trust-ND. It is used to detect fake NA messages. However, the experiments show some limits of this method.

In [13], the authors have presented a technique for detecting Neighbor Solicitation spoofing and advertisement spoofing attacks in IPv6 NDP. However, this method can only detect NS spoofing, NA spoofing, and DoS attacks. The disadvantage of this method is that it does not detect other attacks like Duplicate Address Detection attacks.

In [14], the authors have proposed a new method to secure NDP attacks; this method is based on the digital signature. It detects the messages NS and NA spoofing and DoS attacks, router redirection, and Duplicate Address Detection, but this mechanism is not complete.

In [15], the authors describe and review some of the fundamental attacks on NDP, prevention mechanisms, and current detection mechanisms for NDP-based attacks.

In this paper, we propose to study and evaluate the security in the NDP within the network based on IPv6 protocol. Indeed, we suggest a new algorithm which could secure the attacks in the DAD process. The results showed that DAD process could be optimized by introducing a new field in the NS and NA messages; the hash of the new node's target address. Overall, this method showed a significant effect in terms of time and computation.

3. Features of IPv6 NDP: Duplicate Address Detection

The Neighbor Discovery Protocol (NDP) mechanism provides IPv6 with some number of features essential for the proper IPv6 protocol functioning. The best known is the address resolution feature that matches what is ARP in IPv4. This protocol also offers other features. The one that will interest in our paper, Duplicate Address Detection (DAD), allows detect when two nodes want to use the same address and avoids the future collision by refusing the assignment of the address. This is equivalent to "gratuitous ARP" in IPv4. This feature is even more important, that, in IPv6, new nodes can use the "stateless autoconfiguration" and assign themselves an address (self-generated).

3.1. DAD Process. The Duplicate Address Detection mechanism applies to all type addresses unicast before they are assigned to network interfaces, regardless of whether they are manual, stateless, or stateful. This feature can still be disabled by system administrators.

The Neighbor Discovery Protocol mechanism uses ICMPv6 type messages [2]. Under the DAD mechanism, we are only interested in two types of messages, the Neighbor Solicitation (NS) and the Neighbor Advertisement (NA). When resolving an address, the message Neighbor Solicitation is used to request the physical address of a node (e.g., MAC address) it wants to communicate by contacting it via IPv6 address. This message contains a target field that is populated with the node's IPv6 address that we want to contact. If this target exists, it responds with a message to the request sending node and contains, in one of its fields, an option with the physical address of this node regarding the network interface concerned. This association between the logical address and the physical address will then be kept in the neighbor cache table.

For the DAD mechanism, this request/response exchange is used more finely. The node does not appropriate the address it desires until the procedure has been completed satisfactorily; during this procedure, this address will be called "tentative". To be more precise, if the node receives traffic destined for a "temporary" address, it must not process it or respond to it. The procedure is to issue a Neighbor Solicitation message with as target its "temporary" address and in source address, the address with type "unspecified" (: :). If someone answers, to this NS message with a Neighbor Advertisement message means that the address is already taken and a node already has this address, it is considered that the attempt to obtain an address fails: the node cannot

get this address. There is no other attempt to get this address; the administrator must intervene on the node to configure it with another address. There is another case where we cannot get the address: when the node receives a message NS with as target address the “temporary” address that it wants to use. This means that another node also performs a DAD procedure for the same address. In this case, neither of the two nodes performing the DAD mechanism on the same address will be able to obtain it.

The DAD mechanism is not infallible, especially if it occurs during the time when several nodes of the same network are temporarily “separated” (loss of connecting or dropping a link between the nodes) and that one or more of the nodes perform a DAD procedure. They can assign the same address without the collision detection procedure.

3.2. The Algorithm of DAD Process. For the node, the procedure starts by listening to the multicast group “all-nodes multicast” and the multicast group of the solicited-node (“solicited-node multicast”). The first allows it to receive address resolution requests (“Address Resolution”) for this address and the second will allow it to receive the messages sent by other nodes also making a DAD on this address. In order to listen to these, the node must send a Multicast Listener Discovery (MLD) [16] request; when a node triggers the DAD procedure, it sends a Neighbor Solicitation message, an ICMPv6 type message.

The header IPv6 contains the following fields:

- (i) The source address of the IPv6 packet is the unspecified address (: :).
- (ii) The destination address is the multicast address of the solicited-node (“Solicited-Node Multicast Address”) of the “tentative” address, that is the last three octets of the provisional address concatenated with the prefix FF02:: 1: FF00: 0/104.

When sending this NS message, we observe for the ICMPv6 header:

- (i) The target address field is filled with the “tentative” address.
- (ii) The link layer option of the source is not used. So, two nodes can send the same identical NS message.

With stateless autoconfiguration, it is important to note that if the DAD mechanism fails, then there is no further testing and a new address will have to be assigned otherwise, in particular, for addresses that will have been built automatically via the modified EUI-64 format.

The algorithm DAD is described as follows:

- (i) The first step is to generate an IPv6 address with either autoconfiguration or other methods.
- (ii) In the second step, the node will be subscribed in multicast groups: all multicast nodes and solicited multicast node.
- (iii) After, there are three cases:

- (a) A NA message is received: the tentative address is used as a valid address by another node. The tentative address is not unique and cannot be retained.
- (b) A NS message from a neighbor is received as part of a DAD procedure; the tentative address is also a tentative address for another node. The tentative address cannot be used by any other node.
- (c) Nothing is received after one second (default value): the tentative address is unique, it passes from the provisional state to a valid one, and it is assigned to the interface.

Figure 1 shows the DAD algorithm.

3.3. The Attack on DAD Process. An attack on the DAD mechanism was identified in [10]. The attack is composed as follows: the attacker will deceive the DAD mechanism and make it succeed in one of the two cases where it fails so that the victim cannot claim an address. Since there is a finite number of tries to get an address, the DAD always ends up failing; it is a DoS attack [11]. For the attack to be feasible, the attacker must be able to listen on the network to any query necessary to perform the DAD procedure, e.g., the NS messages with the unspecified address as the source address is characteristics of the DAD procedure; this implies being able to join the multicast group “Solicited-Node”. He then has two choices; he can send a NS message with, as source address, the unspecified address and, as the target address, the address of the victim or an NA message with, as the target address, the “tentative” address of the victim. It can thus prevent the arrival of new nodes having no address yet. The attack effectiveness depends strongly on the type of links, because it is necessary that the attacker can receive the first NS sent by the victim and that he can answer them. Indeed, the attacker must be able to join the multicast group “Solicited-Node”, which is not easy in the case of a level 2 point-to-point technology, for example, ADSL.

3.4. Vulnerabilities of Multicast Communications. In IPv6 multicast (DAD process), groups are identified by a group address and any node in the network can join or leave the group when it wishes. This simplicity, which is the power of multipoint routing, presents however many vulnerabilities such as

- (i) IPv6 multicast does not support the notion of the closed group. Indeed, multicast addresses are public: joining a group or leaving a group is an operation that does not require special permissions. This allows any node to join a group and receive messages for it.
- (ii) Access to the group is not controlled: an intruder can send data to the group without being part of it, disrupt the multipoint session, and possibly cause congestion in the network.
- (iii) The data intended for the group can cross several unsecured channels before reaching all members of

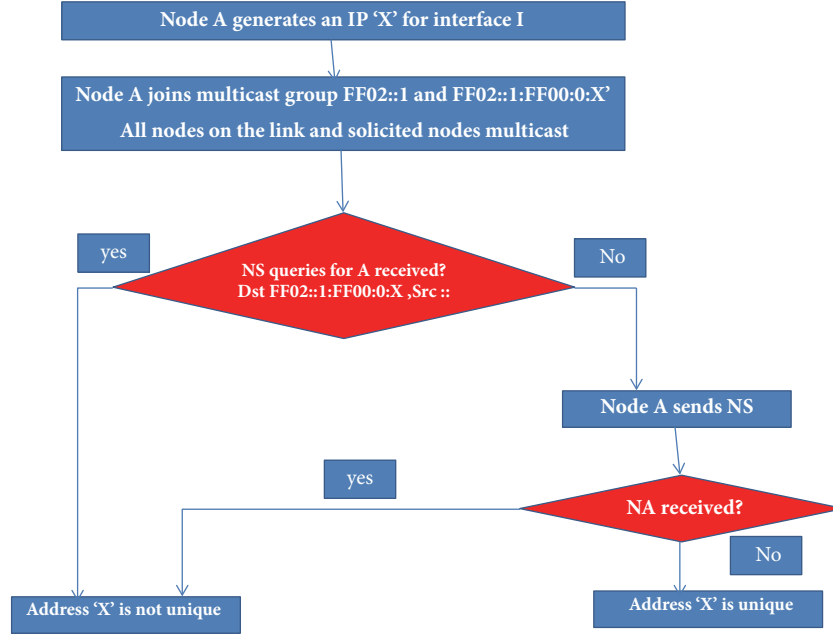


FIGURE 1: The flowchart of the DAD process.

the group. This increases listening opportunities to potential intruders.

- (iv) Group communications offer more opportunities for intercepting communications, proportional to the number of participants.
- (v) A vulnerable point in the group implicates the safety of all members of the group.
- (vi) The large-scale publication of the group's identity and address helps intruders focus their attacks.
- (vii) Attackers can impersonate the legitimate members of the group.

To counteract these attacks, group communication requires security services such as authentication, data privacy, and confidentiality of the traffic flow.

3.5. Security Needs in Multicast. Multicast requires the set of security mechanisms in a unicast communication in addition to some needs inherent to its nature which is the group communication. These needs can be divided into three main parts.

3.5.1. Authentication. All participants in a multicast session must self-authenticate before joining the group. Authentication [17] may be restricted to group members such as sources and receivers or possibly extended to the routing infrastructure like designated routers.

Among other authentication mechanisms, the certification scheme with a third authority can be used.

3.5.2. Integrity. This ability ensures that the multicast stream reaches the recipients without falsification. This option is

usually provided by cryptographic, hash, and digital signature mechanisms [18].

3.5.3. Confidentiality. This confidentiality [19] must be provided at several levels:

- (i) Past privacy (backward confidentiality): we can imagine that a hacker can store the multicast stream for a time interval $[t_0, t]$ and join the group at time t to acquire the keys needed to decrypt this stream "past". Past privacy alters such a hacking scheme by (for example) modifying decryption keys for the stream, once a new member joins the group.
- (ii) Forward confidentiality: a system with this ability prevents any member excluded from the multicast group at time t from having the keys necessary for decrypting the multicast stream at times $t + \mu$. This usually results in a modification of these keys and then their redistribution to the remaining members.
- (iii) Group privacy: only authenticated members must have the keys to decrypt multicast messages.

4. Proposed Algorithm Model

In this section, we present the description of our algorithm which makes it possible to secure the target address used in NS message.

4.1. Digital Signature. A digital signature must prove the identity of the issuer of NS or NA messages and guarantee nonrepudiation. The RSA cryptosystem also allows the signing of a message. Indeed, by inverting the mechanisms that are to say that the decryption of the message, which is only

accessible to those who know the factorization of the module, becomes the signature process. On the other hand, since encryption is public, by encrypting the signature produced, everyone must fall back on the message. The ability to decrypt with RSA proves the knowledge of the private key.

4.2. Hash Function. A hash function [20] is a method for characterizing information, a data. By having a sequence of reproducible treatments at an input, it generates a fingerprint to identify the initial data.

A hash function, therefore, takes as input a message of any size, applies a series of transformations, and reduces this data. We get at the output a string of hexadecimal characters, the condensed, which summarizes somehow the file.

We define a hash function as an application:

$$h : \{0, 1\}^* \rightarrow \{0, 1\}^n, \quad n \in \mathbb{N}. \quad (1)$$

A hash function is considered safe if the following three properties are satisfied:

- (a) Resistance to a preimage attack (one-way). For any given output y , finding an x , which makes $h(x) = y$, is computationally infeasible.
- (b) Resistance to a second preimage attack. For any given input x , finding an input x' that is unequal to x , which makes $h(x) = h(x')$, is computationally infeasible.
- (c) Resistance to a collision attack. Finding two unequal inputs x and x_0 , such that $h(x) = h(x')$, is computationally infeasible.

The SHA Secure Hash Algorithm [21] is a hash algorithm used by certificate authorities to sign certificates and CRL (certificate revocation list). Introduced in 1993 by the NSA with the SHA0, it is used to generate unique condensates (thus for “chopping”) of files.

4.3. The Structure of SHA1 and SHA512. Table 1 shows the characteristics of SHA-1 and SHA-512.

4.4. Hash-TargetAdd-DAD. Hash-TargetAdd-DAD (Hash target address) is a new definition of the ICMPv6 packet (for NS and NA).

Since the standard DAD is not secure, in order to fulfill such security requirement, a “Hash Secure Target” can be applied on NS and NA messages to ensure that only nodes which possess this hash are able to communicate in the IPv6 local network.

Figure 2 shows the message format of Hash-TargetAdd-DAD.

The message format of Hash-TargetAdd-DAD is illustrated in Figure 3 Hash-TargetAdd-DAD uses two new message types, namely, $NS_{hash-targetAdd-DAD}$ and $NA_{hash-targetAdd-DAD}$, and its “Type” fields are 138 and 139, respectively. Compared with the NDP packet, Hash-TargetAdd-DAD adds a new field “Hash_target_64”, which stores the last 64 bits of the SHA-512 result.

The hash_target_64 calculation method is illustrated in Figure 3.

TABLE 1: The characteristics of SHA-1 and SHA-512.

| | SHA1 | SHA512 |
|----------------|-----------------------------|------------------------|
| Message size | 2^{64} bits maximum | 2^{128} bits maximum |
| Block size | 512 bits | 1024 bits |
| Word size | 32 bits | 64 bits |
| Size of digest | 160 bits | 512 bits |
| Security level | Collision in 263 operations | 2^{128} bits |

4.5. Secure Target Generation and Matching Process. Each node including the new one begins by generating two public and private keys. Before sending the message, it will encrypt the NS with the private key and then multicasts its public key. When receiving an encrypted message, the receivers will decrypt the NS message with the received public key.

In the case where the receiving node wants to send an NA message, it will encrypt it with its private key and send the encrypted message and the public key to the new node. The new node will decrypt the NA message with the public key sent.

Public and private keys are generated using the RSA algorithm [22].

Rule. If the secure target is the same, then the NS is authentic; else drop the message.

- (i) NS message sending step:

Before the new node sends the NS message, it proceeds as follows:

- (ii) First, it generates the target address fingerprint using an SHA-512 hash function and it extracts the 64 bits from the result of E_s :

$E_s = \text{SHA-512 (target address)}$, where E_s is target address fingerprint

$E_{64} = \text{Hash_target_64}$.

- (iii) Then, it encrypts E_s with its private key:

Signature (Target address) = $C(E_{64})$, where C is an RSA encryption function using the new node private key.

Figure 4 shows the mechanism of the secure NS message sending encrypted with the private key using RSA signature.

- (i) NS message receiving step:

First, upon receipt of the secure NS, the existing will decrypt it with the public key of the new node (generated by the RSA algorithm). Then, it generates the fingerprint of its IPv6 address, using the same hash function as the new node (SHA-512). Finally, it compares the generated fingerprint and that resulting from the signature.

If both fingerprints are identical, the signature is validated. We are therefore sure that

- (ii) This is the new node that sent the NS message.
- (iii) The NS message has not changed since the new node signed it.

| | |
|-----------------|----------------|
| Ethernet header | Dest MAC |
| | Src MAC |
| | Type |
| IPv6 header | Src address |
| | Dest address |
| | Next header |
| ICMPv6 header | Type |
| | Target address |
| | Options icmpv6 |
| | Hash_target_64 |

FIGURE 2: The message format of Hash-TargetAdd-DAD.

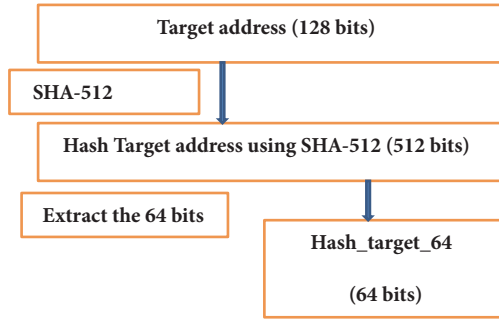


FIGURE 3: Message format of Hash-TargetAdd-DAD.

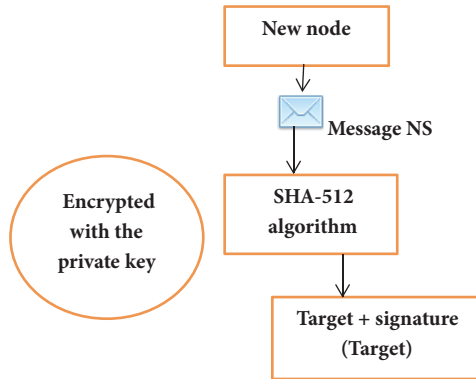


FIGURE 4: The secure NS sent.

In other words, if $D(C(E_{64})) = E_{64}$ (IPv6-existing node)

Where D is an RSA decryption function using the new node public key.

Figure 5 shows the mechanism of the secure NS message receiving decrypted with the public key of the new node using RSA signature.

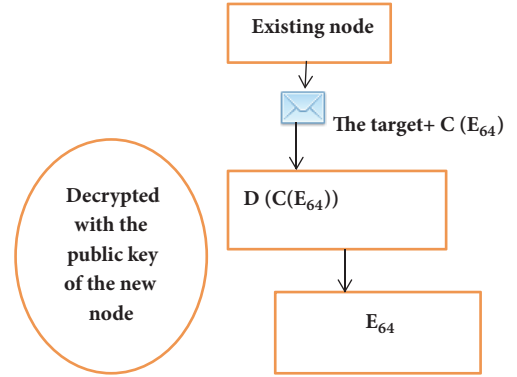


FIGURE 5: The secure NS received.

4.6. *Algorithm HSEC-Target-DAD (HASH Secure Target Address in DAD Process)*. The steps of our algorithm are defined as follows:

- (i) First, the new node generates an IPv6 (the tentative address).
- (ii) The new node uses a hash function to hash the target address with SHA-512.
- (iii) It extracts the last 64 bits from the hash.
- (iv) The hash-64 will be appended to NS message.
- (v) The hash-64 is encrypted by the new node private key and it is sent to multicast address FF02::8 instead of all-nodes solicited multicast group FF02::1 to exclude an attacker who can join the group FF02::1 (all nodes of the network).
- (vi) Existing nodes will decrypt the received NS message with public key new node and match its generated hash secure target with sender's hash secure target.
- (vii) If the sender and a receiver hash secure target match, then the verification of the IP address will take place; otherwise, if no match of hashTag is found the receiver

will discard the NS message and add the MAC address into the blacklist.

- (viii) The existing node will check its IPv6 address with the new node target address if the match of hash target.
- (ix) If the match of duplicate IPv6 address is found, receiving node will reply with NA message appended with the hash target and encrypted by the private key of the existing node. However, if the target address is found unique it creates an entry in its neighbor cache table in order to maintain and update the table for future communication.
- (x) When receiving the NA message, the new node will decrypt it with the public key of existing node. If the match of the hash secure target is found then it performs new DAD process, else it will simply discard the message and add the MAC address to the blacklist.

Our algorithm (Figure 6) is based on target address hashing using the SHA-512 function, then we extract the 64 bits of the result, and we encrypt the NS message with a private key new node; this key is generated from the RSA algorithm [23]. Upon receipt of the secure NS message, the receiver uses the public key new node also generated from the RSA algorithm, to decrypt the received message.

Figure 6 shows the flowchart of our algorithm.

5. Implementation and Evaluation

5.1. Network Topology. The network environment includes a gateway router, an Ethernet switch, a new node (MN1), two existing nodes (MN2 and MN3), and an attacker. Figure 7 shows the network topology. The simulated network is a LAN network.

Each node can have several addresses and centralized random address space to increase the probability of address conflict.

5.2. Simulation Results and Evaluation. In our simulation, we define the following performances:

- (i) Address Configuration Failure Probability (ACFP): when a mobile node uses DAD process to configure its address in the presence of an attack. If a DAD process (DAD-P) is performed y times, and x times have failed, then the ACFP of DAD-P is

$$ACFP = \left(\frac{x}{y} \right) \quad (2)$$

So, since Address Configuration Success Probability (ACSP) is the complement [24] of ACFP then it is defined as follows:

$$ACSP = 1 - \left(\frac{x}{y} \right) \quad (3)$$

From the definition of ACSP, we can conclude that if ACSP is equal to 0, it means that the DAD-P is failed y times,

then the attack is fully functional in DAD-P. Thus, we can use the ACSP to measure a DAD-P.

The simulation includes two scenarios. Scenario 1 simulates DAD and HSEC-Target-DAD with the occurrence of an attack node.

- (i) *Scenario 1:* simulation of DAD and HSEC-Target-DAD with the occurrence of an attack node.
- (a) Results analysis: the simulation results are presented in Figure 8. The results of the simulation show when there is an attacker in the network, with the standard DAD, the configuration of the address generated to the new node fails, which shows that ACSP tends to 0. However, with our algorithm, the attacker cannot decrypt the sent message because he does not have the private key, which shows that ACSP tends to 1.

We can see in the figure that ACSP of HSEC-Target-DAD is higher than DAD.

- (ii) *Scenario 2:* simulation of pseudocollision attacks and SLAAC attacks against HSEC-Target-DAD.
- (a) Pseudocollision attacks: before using a pseudocollision attack [25], we first need to define how a hash function internally works.

Most hash functions are basically composed out of four functions:

- (i) The first function is called an initialization function; it just sets a bunch of start values for the state: $I: \emptyset \rightarrow \{0, 1\}^k$
- (ii) The second function is called an input preprocessing function; it computes some values based on the message and possibly hidden context: $P: \{0, 1\}^l \rightarrow \{0, 1\}^q$
- (iii) The third function is called a state-update function, sometimes also called “compression function”; it takes the current message block, the associated preprocessing, and the current state and outputs a new state: $U: \{0, 1\}^l \times \{0, 1\}^q \times \{0, 1\}^k \rightarrow \{0, 1\}^k$
- (iv) The fourth function is called an output function; it takes the state and outputs the hash digest: $O: \{0, 1\}^k \rightarrow \{0, 1\}^o$

Now a normal collision attack takes the standard composition of these functions and tries to find a collision.

A pseudocollision attack on the other hand just tries to find a collision on the state-update function. So an attacker is interested in finding two triples $x = (m, p, h)$, $x' = (m', p', h')$ such that $U(x) = U(x')$ with $x \neq x'$.

Pseudocollision attacks: this method attempts to search for one or more collision addresses (the IPv6 address with a hash value whose last 64 bits are the same as that in the “Hash_target-64” field) after the attack node receives NS. Then, a number of $NA_{hash_targetAdd-DAD}$ is sent to increase the probability of a successful attack.

SLAAC: in SLAAC attack [26], the node can obtain an IPv6 address by combining its own MAC address and network prefix according to “EUI-64.” Thus, the attack node can

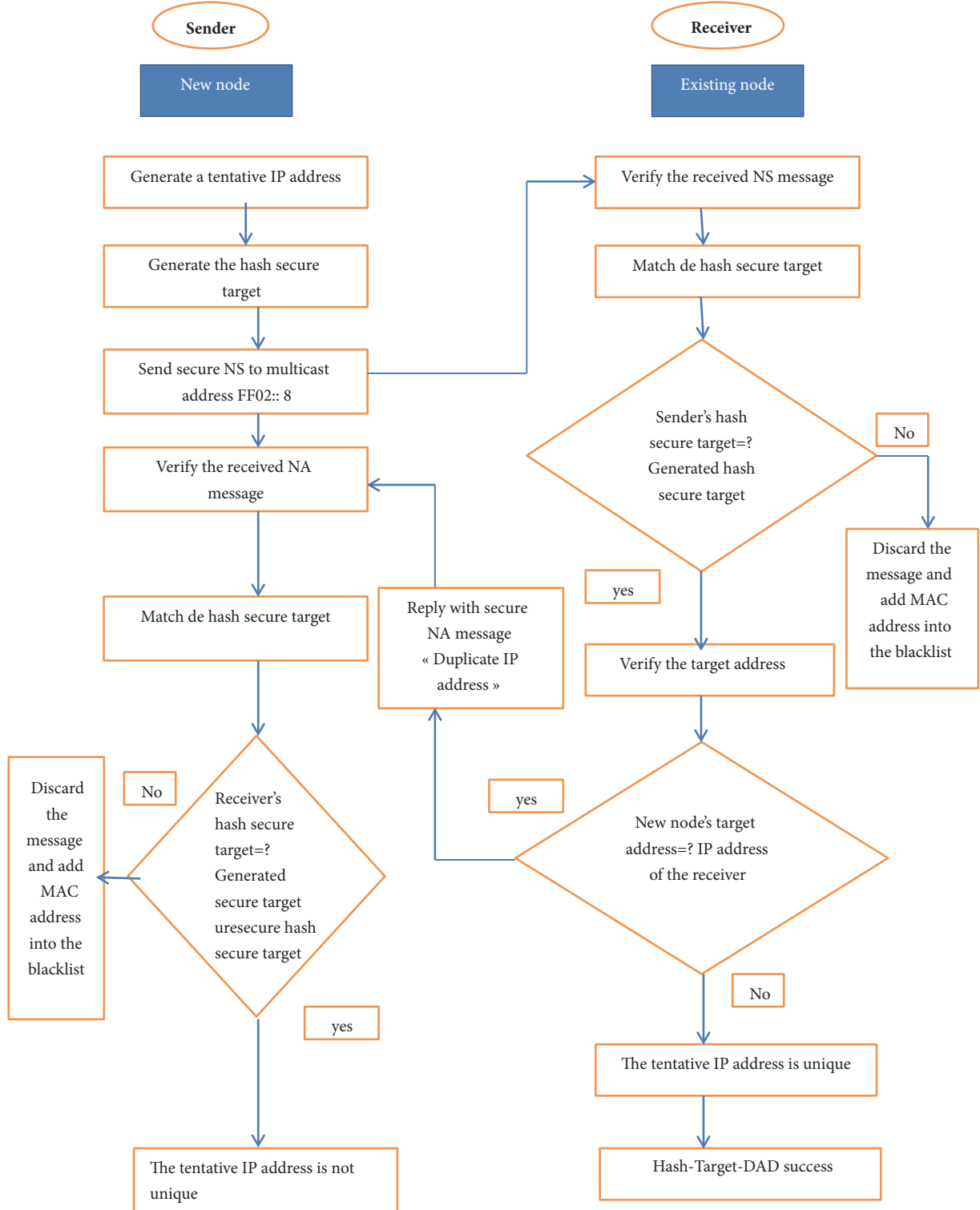


FIGURE 6: The flowchart of the proposed algorithm.

use the characteristics of SLAAC by combining the network prefix and source MAC address in the $NS_{hash-targetAdd-DAD}$ to infer the destination address of DAD.

We set DAD process to 10 seconds. The simulation results are shown in Figure 9. For pseudocollision attack, although

the address space is 2^{32} and the attack node has 10 seconds to seek all collisions, the preimage is difficult to find by the attacker as shown in Figure 9. For a SLAAC attack, the address is formed by EUI-64 method [27]. Using this method, the attack node can attack the network; thus, the ACSP is

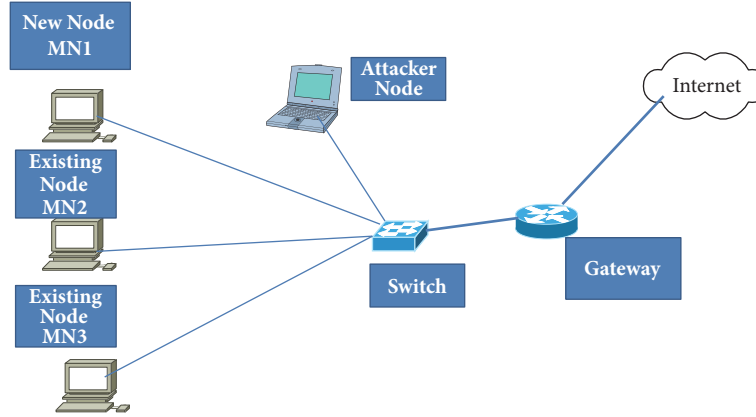


FIGURE 7: The network topology.

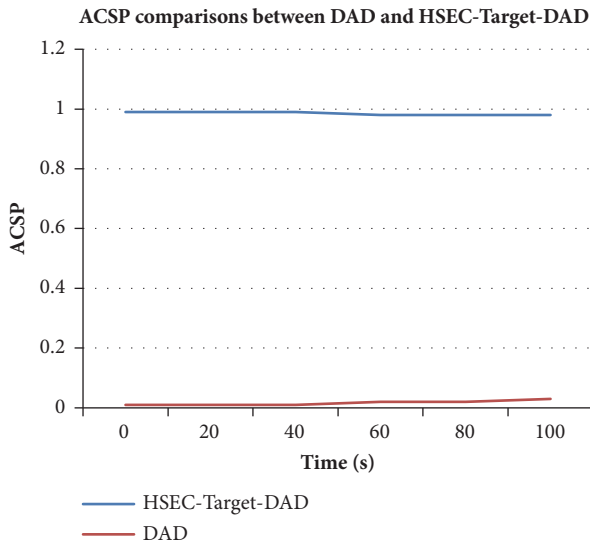


FIGURE 8: ACSP comparisons between DAD and HSEC-Target-DAD.

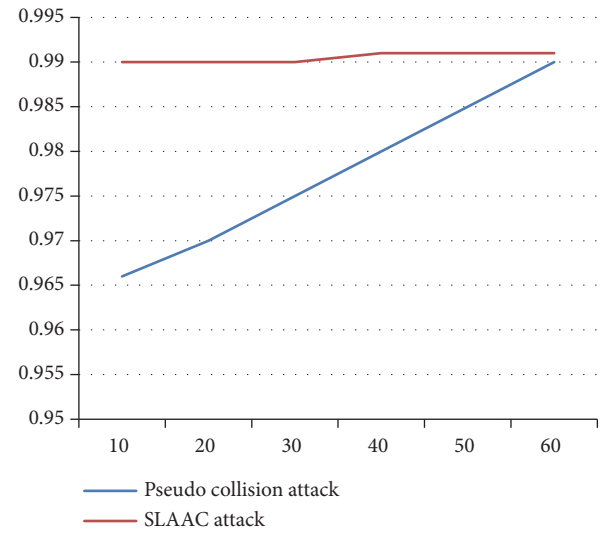


FIGURE 9: ACSP comparisons between pseudocollision attack and SLAAC attack.

considerably low during the early stage of the simulation. Then, the implementation of the blacklist in the algorithm will have its effect.

When DAD process is failed, the construction of the ID (64 bits) is done randomly. Thus, SLAAC attack does not work anymore, and the ACSP of the subsequent HSEC-Target-DAD process gradually increases and approaches to the ACSP of pseudocollision attack.

The effectiveness of our algorithm:

- (i) CGA use SHA1 as a hash function; however in this paper, we use SHA512.
- (ii) SHA512 is faster when the size of input data is large, in our case; the size of the target address is 128 bits.
- (iii) Another effectiveness of our algorithm is that it uses asymmetric encryption to sign messages.
- (iv) Hash_target_64 field can effectively prevent attacks.

6. Conclusion and Perspectives

In order to ensure that all configured addresses are likely to be unique on a given IPv6 link, the nodes execute a Duplicate Address Detection algorithm. Nodes must execute the algorithm before assigning addresses to an interface.

For security reasons, the uniqueness of all addresses must be verified prior to their assignment to an interface. The situation is different for addresses created by stateless automatic configuration. The uniqueness of an address is determined primarily by the portion of the address formed from an interface ID. Therefore, if a node has already verified the uniqueness of a link-local address, we do not need to test the additional addresses individually. The addresses must be created from the same interface ID. All manually obtained addresses must be individually tested to ensure their uniqueness. System administrators at some sites believe that the benefits of Duplicate Address Detection are not worth the overhead they use. For these sites, the use of Duplicate

Address Detection can be disabled by setting an interface configuration flag.

In this paper, we have developed a new algorithm to secure the DAD process in IPv6 network for the small objects in an IPv6 network. This method is based on the security of NS and NA messages. First, before sending the NS message, the new node uses the hash function SHA-512 to hash to the target address and extracts the last 64 bits and then encrypts the result with the public key sent by the initiator of the multicast group FF02::8. When receiving the secure message, the existing nodes decrypt it with its private key.

Then, a hash check must be done; if the hashes are the same, the verification of the IP addresses can be done; otherwise, the message will be deleted.

The underlying cryptosystem, used to generate the public and private key, is RSA algorithm. We used this algorithm for signing the sent message.

The simulation results show that our algorithm has a higher Address Configuration Success Probability than the standard DAD process.

Although IPv6 node communications are limited to NDP and DAD protocols when IPv6 is not officially deployed, there are still attacks that can affect network performance by exploiting only these two protocols as we have been able to study. Our future work will be focalized on router discovery security.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," RFC Editor RFC8200, 2017.
- [2] A. S. A. M. S. Ahmed, R. Hassan, and N. E. Othman, "IPv6 neighbor discovery protocol specifications, threats and countermeasures: a survey," *IEEE Access*, vol. 5, pp. 18187–18210, 2017.
- [3] F. Gont, A. Cooper, D. Thaler, and W. Liu, "Recommendation on stable IPv6 interface identifiers," RFC Editor RFC8064, 2017.
- [4] F. Alisherov and T. Kim, "Duplicate address detection table in IPv6 mobile networks," in *Advanced Communication and Networking*, C. C. Chang, T. Vasilakos, P. Das, T. Kim, B. H. Kang, and M. K. Khan, Eds., vol. 77 of *Communications in Computer and Information Science*, pp. 109–115, Springer Berlin Heidelberg, Berlin, Germany, 2010.
- [5] M. Moslehpour and S. Khorsandi, "A distributed cryptographically generated address computing algorithm for secure neighbor discovery protocol in IPv6," *International Journal of Computer and Information Engineering*, vol. 10, no. N6, 2016.
- [6] C. Dobraunig, M. Eichlseder, and F. Mendel, "Analysis of SHA-512/224 and SHA-512/256," in *Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security*, Advances in Cryptology – ASIACRYPT 2015, Auckland, New Zealand.
- [7] J. L. Shah and J. Parvez, "IPv6 cryptographically generated address: analysis and optimization," in *Proceedings of the AICTC '16 Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, vol. 13, 2016.
- [8] J. L. Shah and J. Parvez, "Optimizing security and address configuration in IPv6 SLAAC," in *Proceedings of the 11th International Conference on Communication Networks, ICCN 2015*, pp. 177–185, ind, August 2015.
- [9] J. L. Shah, "A novel approach for securing IPv6 link local communication," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 136–150, 2016.
- [10] X. Wang, Y. Mu, G. Han, and D. Le, "A secure IPv6 address configuration protocol for vehicular networks," *Wireless Personal Communications*, vol. 79, no. 1, pp. 721–744, 2014.
- [11] Y. Lu, M. Wang, and P. Huang, "An SDN-based authentication mechanism for securing neighbor discovery protocol in IPv6," *Security and Communication Networks*, vol. 2017, pp. 1–9, 2017.
- [12] S. Praptodiyono, I. H. Hasbullah, M. M. Kadhum, R. K. Murugesan, C. Y. Wey, and A. Osman, "Improving security of duplicate address detection on IPv6 local network in public area," in *Proceedings of the 2015 9th Asia Modelling Symposium (AMS)*, pp. 123–128, Kuala Lumpur, Malaysia, September 2015.
- [13] F. A. Barbhuiya, G. Bansal, N. Kumar, S. Biswas, and S. Nandi, "Detection of neighbor discovery protocol based attacks in IPv6 network," *Networking Science*, vol. 2, no. 3-4, pp. 91–113, 2013.
- [14] R. Hassan, A. S. Ahmed, and N. E. Osman, "Enhancing security for IPv6 neighbor discovery protocol using cryptography," *American Journal of Applied Sciences*, vol. 11, no. 9, pp. 1472–1479, 2014.
- [15] M. Anbar, R. Abdullah, R. M. A. Saad, E. Alomari, and S. Alsalem, "Review of security vulnerabilities in the IPv6 neighbor discovery protocol," *Lecture Notes in Electrical Engineering*, vol. 376, pp. 603–612, 2016.
- [16] Sridevi, "Implementation of multicast routing on IPv4 and IPv6 networks," *International Journal on Recent and Innovation Trends in Computing and Communication*, pp. 1455–1467, 2017.
- [17] Y. Cunjiang, X. Dawei, and J. Li, "Authentication analysis in an IPV6-based environment," in *Proceedings of the 2013 3rd International Conference on Computer Science and Network Technology*, 2013.
- [18] M. A. Nia, A. Sajedi, and A. Jamshidpey, "An introduction to digital signature schemes," in *Proceedings of the Telecommunications (IST)*, 2014.
- [19] K. Chittimaneni, M. Kaeo, and M. Kaeo, "Operational security considerations for IPv6 networks," *Internet-Draft*, 2014.
- [20] N. Abdoun, S. El Assad, M. A. Taha, R. Assaf, O. Deforges, and M. Khalil, "Secure hash algorithm based on efficient chaotic neural network," in *Proceedings of the 2016 International Conference on Communications (COMM)*, pp. 405–410, Bucharest, Romania, June 2016.
- [21] S. Gupta, N. Goyal, and K. Aggarwal, "A review of comparative study of MD5 and SSH security algorithm," *International Journal of Computer Applications*, vol. 104, no. 14, pp. 1–4, 2014.
- [22] Saranya, Vinothini, and Vasumathi, "A study on RSA algorithm for cryptography," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5708–5709, 2014.
- [23] R. Pir, "Security improvement and speed monitoring of RSA algorithm," *IJEDR*, vol. 4, no. 1, 2016.
- [24] "Complementary event," https://en.wikipedia.org/wiki/Complementary_event.
- [25] G. Wang and Y. Shen, "Preimage and pseudo-collision attacks on step-reduced SM3 hash function," *Information Processing Letters*, vol. 113, no. 8, pp. 301–306, 2013.
- [26] F. J. Buenaventura, J. P. Gonzales, M. E. Lu, and A. V. Ong, "IPv6 stateless address autoconfiguration (SLAAC) attacks and

detection,” in *Proceedings of the DLSU Research Congress*, vol. 3, 2015.

- [27] P. Tayal, “IPv6 SLAAC related security issues and removal of those security issues,” *International Journal of Engineering and Computer Science*, vol. 3, no. 9, pp. 8445–8459, 2014.

Research Article

A Feasible Fuzzy-Extended Attribute-Based Access Control Technique

Yang Xu ¹, Wuqiang Gao,¹ Quanrun Zeng,¹ Guojun Wang,² Ju Ren,¹ and Yaoxue Zhang¹

¹*School of Information Science and Engineering, Central South University, Changsha 410083, China*

²*School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Yang Xu; xuyangcsu@csu.edu.cn

Received 29 December 2017; Revised 19 April 2018; Accepted 29 April 2018; Published 5 June 2018

Academic Editor: Debasis Giri

Copyright © 2018 Yang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Attribute-based access control (ABAC) is a maturing authorization technique with outstanding expressiveness and scalability, which shows its overwhelmingly competitive advantage, especially in complicated dynamic environments. Unfortunately, the absence of a flexible exceptional approval mechanism in ABAC impairs the resource usability and business time efficiency in current practice, which could limit its growth. In this paper, we propose a feasible fuzzy-extended ABAC (FBAC) technique to improve the flexibility in urgent exceptional authorizations and thereby improving the resource usability and business timeliness. We use the fuzzy assessment mechanism to evaluate the policy-matching degrees of the requests that do not comply with policies, so that the system can make special approval decisions accordingly to achieve unattended exceptional authorizations. We also designed an auxiliary credit mechanism accompanied by periodic credit adjustment auditing to regulate expediential authorizations for mitigating risks. Theoretical analyses and experimental evaluations show that the FBAC approach enhances resource immediacy and usability with controllable risk.

1. Introduction

The burgeoning communication and computing technologies such as the 5G mobile Internet [1] and network computing [2–5] have substantially enhanced the availability and usability of resources to end users. Consequently, new evolutions including the popularity of telecommuting [6] and the general acceptance of “bring your own device” [7] have inadvertently driven the emergence of more complex and diverse resource access and usage scenarios. However, the developments in access control technologies have somewhat lagged behind. The typical role-based access control (RBAC) [8] model and older paradigms such as mandatory access control (MAC) [9] and discretionary access control (DAC) [10] are insufficient to support dynamic, distributed, and unpredictable access scenarios, because of their inherent limitations in flexibility, scalability, adaptability, and control granularity. More effective solutions that consider additional relevant parameters (e.g., subject states, object states, and contextual information) have also been explored, among which the attribute-based

access control (ABAC) is the most promising approach for the new era. It has successfully transitioned from purely academic studies [11–20] to the practical application phase [21–24]. By enforcing attribute-formed policies on access requests, this adjustive, expressive, and highly extensible authorization model has an overwhelmingly competitive advantage, especially in dynamic and complicated environments.

Unfortunately, the ABAC ineluctably encounters practical problems during the use in current dynamic and complex scenarios spawned by the latest communication and computing techniques. Due to the rigid policy-based access control enforcement and the inability to automatically and efficiently handle exceptional access requests, some urgent requests which may not fully comply with the original ABAC policies would not be authorized in time due to the requirements of inefficient human involved approval processes, which impacts the resource availability and thereby affects the business timeliness and even leads to irreversible unfortunate consequences. There is a particular negative example that a world's top chip manufacturer once restricted its private cloud

services only accessible by on-site staffs within the working hours for security purpose. Nevertheless, the staffs were easily frustrated in policy matching due to not only human factors but also technical reasons (The mobile positioning can be unsteady or outdated due to the functional defects or optimization reasons. Besides, the time limit obstructs lots of workflows in practice.). In absence of a flexible and efficient exceptional request handling mechanism, consequently, the working efficiency was severely affected as staffs could not get expected services in time when inefficient administrator involvements were often required for handling exceptional requests. Undoubtedly, the problem can be even worse in some time-sensitive cases, such as the sudden and urgent needs for classified information in stock or futures markets, the remote patient privacy data requirements in emergency surgeries, and the interorganizational confidential information requests in critical intelligence analyses.

Obviously, a more flexible and efficient exceptional access authorization method is badly needed by the stock ABAC paradigm to guarantee the business timeliness, especially for emergency situations, so as to make the ABAC more feasible, flexible, and adaptive for fitting current dynamic, distributed, unexpected, and complicated situations.

In a sense, access control can be regarded as risk control. Therefore, the concept of risk and the opposite concept of trust have naturally been introduced as an effective and flexible assistive tool for the authorization decision-making process. For instance, the risk assessment method has already been integrated into classical models like RBAC and multiple levels of security (MLS) [25, 26]. By estimating the risk of the certain request based on the specific involving information and comparing the risk with some preset acceptance criteria of risk, these risk-oriented enhanced models have achieved flexible and efficient unattended authorizations for urgent requests which do not comply with the basic access rules in original models. More recently, risk and trust evaluation schemes are increasingly viable in access control when taking more parameters (e.g., environment states) into account, which yields more expressive and flexible solutions [27–31]. Because of these encouraging attempts, we are reasonably confident that the ABAC paradigm will benefit from risk evaluation schemes as well, especially the more flexible and efficient decision-making ability to deal with exceptional urgent access requests in dynamic and complex access environments. In this context, fuzzy logic [32], as one of the most recognized math tools for assessment that reasons probability from vague knowledge, is a viable option to determine the semantic matching degree of access requests and ABAC policies.

Focusing on the situations described above, in this article, we propose a feasible ABAC-based access control paradigm named fuzzy-extended ABAC (FBAC) to improve the flexibility and time efficiency when tackling low-risk exceptional authorizations for the emergency cases. We use the fuzzy assessment mechanism to evaluate the policy-matching degrees of requests failed to meet policies and then make authorization decisions according to both the denial threshold and the credit available to the requesters, to achieve unattended temporary authorization for the exceptional

urgent access requests which are initiated by reputable users (reflected by credit values) but slightly violate the predefined ABAC policies. Furthermore, we designed an auxiliary credit system to impose restrictions on special authorizations and perform periodic credit adjustment auditing, to reduce the potential for abuse of expediential approvals. In addition, we describe a detailed case study to help readers understand the FBAC better and finally demonstrate our improvements from the perspectives of usability, security, and performance theoretically and experimentally.

The major contributions of our work are summarized as follows.

(1) We introduce the matching-degree-based fuzzy evaluation method into the original ABAC paradigm, which enables more efficient and flexible unattended approval for exceptional urgent authorization cases, to increase the resource usability and thereby the business timeliness.

(2) We keep the risk of special authorization abuse under control by not only using the configurable threshold to intercept high-risk requests directly but also by building a credit system combined with periodic credit adjustment audit mechanism.

(3) We analyzed the FBAC model theoretically for its usability, risk, and complexity and then implemented a prototype system to evaluate its effectiveness and efficiency by experiments, to demonstrate our enhancements in usability and immediacy, as well as the acceptance of security risks.

The remainder of this article is organized as follows. We introduce some articles related to our work in Section 2. In Section 3, we review several basic concepts of fuzzy logic. In Section 4, we propose our fuzzy-extended ABAC (FBAC) paradigm and detail it in the case study. Section 5 gives a brief discussion of FBAC's usability, risk, and complexity. Then in Section 6, we evaluate our prototype and analyze the experimental results. The last section summarizes this paper and describes possible improvements.

2. Related Work

Access control is an indispensable security technology for preventing sensitive resources from illegal access. A variety of access control models have been studied over the years, and different ones are designed for addressing discrete challenges focusing on confidentiality, integrity, scalability, manageability, etc. Some typical patterns like DAC [10], MAC [9], and RBAC [8] have emerged. Nonetheless, these classical models above are not expressive enough to take into account the effects of other additional factors (e.g., time of the day or user IP). As a result, they are gradually unable to meet the new requirements of geographical, temporal, and context-aware information systems.

Breaking the limitation of the subject-object pattern, more revealing access control paradigms are well studied.

One inspiring endeavor is bringing in risk factor to strike balance between system security and usability. The concept of “fuzzy” has been introduced to the RBAC for achieving better flexibility in handling exceptional requests [25]. The fuzzy RBAC carried out the more relaxed assignments of

user-role and role-permission compared with the original RBAC model. And the assignment degrees were subjectively assigned to represent the accompanying uncertainties and risks of corresponding assignments. Then the access control enforcement was based on the risks of requests reflected by the overall assignment degrees. However, this conceptual solution did not provide a practical and detailed calculation method of assignment degrees. Cheng et al. [26] proposed the fuzzy MLS, a risk self-adjusting access control technique, which can quantify the potential risks associated with the exceptional access and thereby optimize the risk-benefit trade-off. In this model, the risk of the request was quantitatively assessed according to both the value of the object and the empirical illegal disclosure probability determined by the MLS tags (security level, etc.) of the involving subject and object and then made the access decision by comparing the risk with a preset risk scale and asking the user to provide corresponding risk tokens assigned by the administrator. Meanwhile, trust mechanism, closely connected to the concept of risk, has also been ushered in. Dimmock et al. expanded the existing access control framework and combined the trust-based assessment with reasoning to form a dynamic model that can manage risk more intelligently [27]. Liu combined the dynamic hierarchical fuzzy system with trust evaluation, then introduced a fuzzy multiattribute trust access control scheme for cloud manufacturing system [28]. Mahalle et al. [29] developed a trust-extended fuzzy authorization scheme and put forward the concept of trust rating for identity management. Context awareness is a significant precondition for accurately perceiving and properly handling risks. Feng et al. [30] integrated user behaviors and operating environment to propose a scalable trust-based and context-aware access control technique for large-scale, widely distributed networks. Taking into account both factors of trust and environmental perception, Bhatti et al. [31] constructed a trust-enhanced, environment-sensitive authorization model for network traffic based on X-GTRBAC (XML-based generalized temporal RBAC) framework.

As cross-organizational, multisectoral cooperations become integral parts of current business processes, to overcome the drawbacks of the mainstream access control models while unifying their advantages, there has been considerable interest in a more general model, namely ABAC [11, 12], which is considered as “next generation” authorization model for its dynamic, context-aware, and fine-grained features, defines a multidimensional access control paradigm where access requests are accepted or rejected based on all kinds of assigned attributes, including subject attributes (e.g., age, department, job title), action attributes (e.g., read, write, append), object attributes (e.g., owner, size, classification), and contextual attributes (e.g., time, location), and a set of policies. ABAC empowers more precise access control, facilitating the generation of expressive and flexible policies through the combination of a wide range of factors.

Determined attempts have been made not only by standards organizations [11] but also by many IT giants such as IBM and Cisco [21, 22], which contributes much to the development and widespread deployment of ABAC technique. Meanwhile, the academic community has also invested

significant effort in this research area [13]. Li et al. [14] conducted in-depth discussions on the inherent logical relations and system architecture of ABAC. Jin [15] has formalized the ABAC scheme and achieved the simulation of other classical models. Sookhak et al. [16] carried out an exhaustive survey on ABAC techniques befitting cloud and distributed environment. Based on the authorization requirements of grid systems, Bo et al. [17] developed an efficient multipolicy ABAC technique suitable for grid computing based on the third-party authorization framework.

Regardless the benefits of ABAC, its rigid policy-enforcement mechanism as well as the guideless policy-configuration process may somehow lead to the reduction of resource usability and then the time efficiency of business. Demchenko and Ngo [18] mitigated this problem by proposing a specific ABAC solution for the cloud tenants which enables hierarchical delegations to support the efficient collaborations among tenants. Although this approach contributes to yield a more flexible ABAC paradigm, it is not a general solution which can only fit for limited scenarios. In a more intrinsic view, it reflects the fact that ABAC is thoughtless in how to efficiently deal with exceptional access requests.

Considering all these challenges and even more complex and urgent application scenarios, in our previous conference paper [19], we put forward a rough fuzzy ABAC framework conceptually aiming to achieve flexible special authorizations for exceptional urgent requests with low risks. However, it did not consider the effects of benign users’ unintentional misoperations and ignored the differences in importance among attributes. Besides, its credit management mechanism is not reasonable enough while the experimental evaluation and analysis are not included. This research is inclined to make up for the past deficiencies so as to achieve an innovative approach with the auxiliary exceptional requests handling functionality, for enhancing the resource usability and thereby business timeliness in highly dynamic and unexpectable environments.

3. Preliminary

This section goes through some necessary concepts of the fuzzy theory [32].

Fuzzy Set. Fuzzy set is an extension of sets whose elements have degrees of membership. A fuzzy set can be defined as a pair (U, μ) in which U is the universe set of elements and μ is the membership function that mapping elements to corresponding membership degree, as follows:

$$x \in U \longrightarrow \mu(x) \in [0, 1]. \quad (1)$$

Fuzzy Logic. The fuzzy logic is one type of multivalued logic which is based on fuzzy set theory. In fuzzy logic, the true/false value is replaced with membership values, which are real numbers between 0 and 1. A possible definition of operations in fuzzy logic is based on max/min function [33] in which the AND operator means taking the minimum value among membership values, while the OR operator means taking the maximum.

TABLE 1: The major notations and definitions.

| Notations | Definitions |
|------------------|---|
| q_i | the i th request. |
| p_j | the j th clause in policy set. |
| $a_{j,k}$ | the k th attribute involved in the clause p_j . |
| $w_{j,k}$ | the weight of $a_{j,k}$ in the p_j . |
| $\xi_{j,k}(q_i)$ | The fuzzy membership function for calculating the membership degree of the q_i to the constraint range of attribute $a_{j,k}$. |
| $\nu_j(q_i)$ | The fuzzy membership function for calculating the membership degree of the q_i to the clause p_j . |
| $\mu(q_i)$ | The fuzzy membership function for calculating the membership degree of the q_i to the policies. |
| $cost(q_i)$ | The credit cost of special approval for the q_i . |
| H | The rejection threshold (a rational number in $(0, 1)$). |
| c_{max} | The credit-line (a rational number in $(0, 1)$). |
| c_x | The credit value of the subject x (a rational number in $(0, c_{max})$). |
| r | The credit recover ratio (a rational number in $(0, 1)$). |

4. FBAC

In this section, we define several necessary notations at the beginning. Then, we introduce the architecture of FBAC briefly and describe its workflow step by step. Further, we demonstrate its essential components in detail. And finally, we study a detailed case to help readers understand the FBAC better.

For convenience, we only adopt granting policies (Although the policies in ABAC can be granting or denying ones, they are mutually transformable.) in this paper and employ a refusal precedence principle for the decision-making process; i.e., a granted decision would be made when the request meets at least one clause in the policy set.

4.1. FBAC Model. The FBAC model wraps the standard ABAC as a preliminary screening module and integrates additional decision support components for improving the resource usability, thereby gaining better business timeliness.

Notations. Throughout this paper, we use the notations in Table 1 for simplified description purpose.

Architecture and Workflow. As seen in Figure 1, the FBAC is built upon the standard ABAC model with additional fuzzy evaluation component and credit component. The first component is developed to support unattended special authorizations, while the second is a security remedial measure. These additional components are independent to standard ABAC which contributes to the effortless integration.

When a request is reached, the FBAC firstly collects the states of related attributes of that request, including the attributes of subject, object, context, and action (Steps 1-2). After applying the policies, if this request is not granted by the standard ABAC process, it will be delivered to our fuzzy evaluation component for a further decision based on the membership degree calculation and the rejection threshold filter (Step 3). The credit component will check the available credits of the requester and denies the request if the requester is unable to afford the credit cost for approving this

Input: q_i, c_x
Output: $Decision \in \{\text{granted}, \text{denied}\}$
(1) **if** match any policy **then**
(2) **return granted**
(3) **end if**
(4) $\mu(q_i) \leftarrow \max_{i=1}^n (\nu_i(q_i))$
(5) $cost(q_i) \leftarrow 1 - \mu(q_i)$
(6) **if** $\mu(q_i) < H$ **or** $c_x < cost(q_i)$ **then**
(7) **return denied**
(8) **end if**
(9) $c_x \leftarrow (c_x - cost(q_i))$
(10) **return granted**

ALGORITHM 1: The FBAC Decision-Making Procedure.

exceptional request (Step 4). If the corresponding subject has sufficient credits to pay the credit cost, the credit component will issue a prompt to ask the requester to confirm the credit consumption (Step 5). Once confirmed by the requester, the request will be granted and logged, at the expense of corresponding credit consumption. Note that part of the consumed credit will be restored after audit if the subject is not malicious. Otherwise, this request will be denied (Step 6). The final decision is delivered to the enforcement facility which will mediate the corresponding access to the object accordingly (Step 7). The major decision-making process is illustrated in Algorithm 1.

Apart from the major decision-making process, there is an audit process which will router the recorded exceptional access authorizations to administrators for review periodically. And then the credit audit system will restore a part of the users' credit according to the auditing results (Step a).

Fuzzy Evaluation Component. When a request q_i is rejected by the standard ABAC module because it can not exactly match any policy, the FBAC system will turn to fuzzy evaluation component for further judgments. This component will evaluate the matching degree of the q_i to policies through membership degree calculation. Specifically, for the

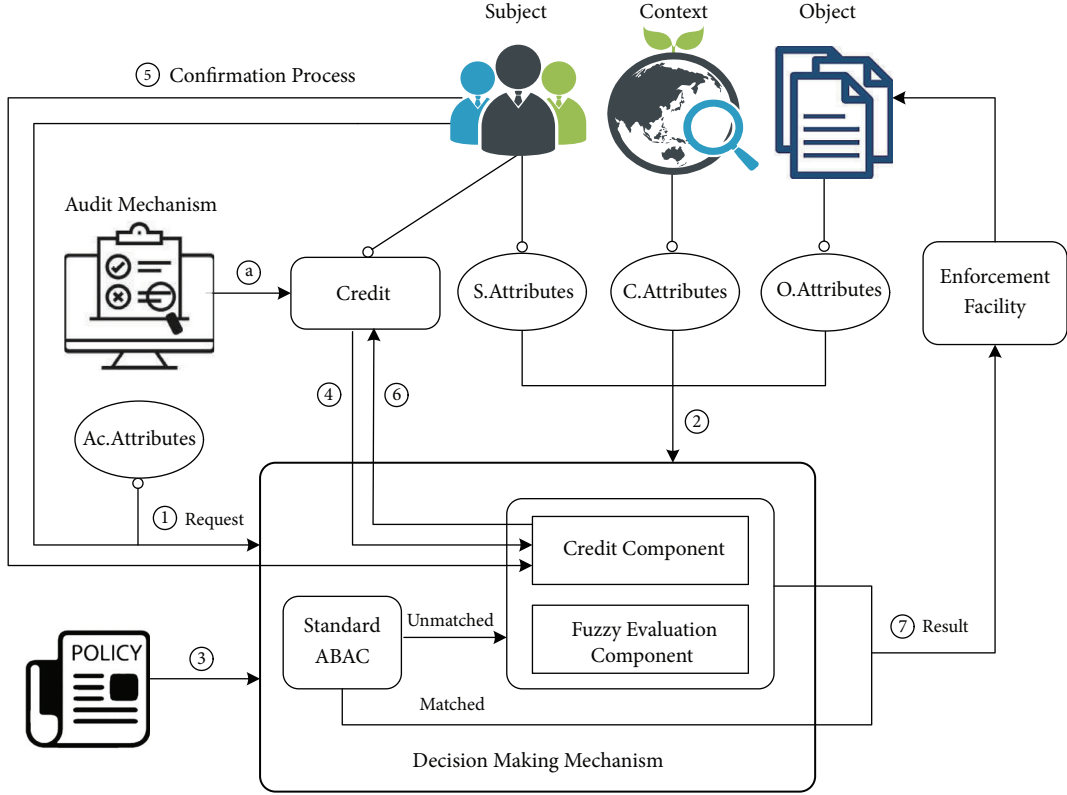


FIGURE 1: Architecture and workflow of FBAC.

j th clause in the policy set, this component will calculate the membership degree of the request q_i to that clause as follows:

$$\nu_j(q_i) = \frac{\sum_{k=1}^n w_{j,k} \xi_{j,k}(q_i)}{\sum_{k=1}^n w_{j,k}}. \quad (2)$$

In formula (2), $\xi_{j,k}(q_i)$ is the membership subfunction that maps q_i to a certain membership degree according to the matching degree of q_i to the constraint range of the k th attribute in the j th clause. The design of $\xi_{j,k}$ is closely related to the meaning of the corresponding attribute and policy clause and also depends on administrators subjectively. There exist several primary guidelines for determining the membership subfunction [34]. And the most commonly recommended function templates include the trapezoid subordinate function, the trigonometric membership function, the step function, etc. In this paper, we select the trapezoid subordinate function and the step function for different policy clauses respectively (cf. Section 4.2). The FBAC gives the administrators greater freedom to determine the attributes which should be fuzzy processed based on practical administrative needs. In general, the continuous attributes can be fuzzy processed, while the discrete ones (e.g., users names) should be fully matched for obtaining final authorizations. Additionally, if the discrete attributes can be somehow transformed into continuous ones based on partial ordered relations, they can also be fuzzy processed similarly, e.g., converting the discrete and hierarchical job titles to continuous level numbers. $w_{j,k}$ is the weight of the

corresponding attribute. Introducing weight factor enables administrators to adjust the influence of each attribute in the policies, so as to provide more flexible and expressive manageability.

Since there usually exist more than one clause in the policy set, the holistic matching degree is synthesized with maximum synthesis rules [33], as shown in the following formula:

$$\mu(q_i) = \max_{j=1}^n \nu_j(q_i). \quad (3)$$

After obtaining the matching degree $\mu(q_i)$, the FBAC will compare $\mu(q_i)$ with the rejection threshold H . If $\mu(q_i) < H$, the request q_i will be denied by FBAC. Otherwise, the credit component will be invoked for supporting further judgments.

Credit Component and Audit Mechanism. The fuzzy evaluation component provides users with extra access opportunities without manual reviews. However, in spite of the benefits in the resource usability and business timeliness, this fuzzy evaluation module poses potential threats such as abuse issues unintentionally. Therefore, we build a credit component combined with periodic credit adjustment auditing mechanism as the countermeasure to mitigate the risk of abuse.

Our credit component maintains a credit value c_{x_s} ($c_{x_s} \in [0, c_{max}]$, where $c_{max} \in (0, 1)$ is the preset credit line) for each subject x_s . When the FBAC is initialized, every c_{x_s} will be set as c_{max} without discrimination. During the use, the

credit component will be invoked to provide further decision support for the request q_i if its matching degree $\mu(q_i)$ exceeds the rejection threshold H . We define $cost(q_i) = 1 - \mu(q_i)$ as the special approval cost for the request q_i with the matching degree $\mu(q_i)$, because the $cost(q_i)$ can reflect the gap between the states of the q_i and the precise requirements of policies. Thus, the credit component will compare the credit c_x of the requester x with the corresponding special approval cost $cost(q_i)$. If $c_x < cost(q_i)$, then a denial suggestion will be issued for the q_i as the requester does not have enough credit to afford the cost. Otherwise, the FBAC will ask the requester for confirmation to consume that $cost(q_i)$ and enforce the requester to comment reasons for the unusual request. This additional prompt scheme is quite useful to avoid user misuse and is also helpful for future audits. Then if the requester x replies in the affirmative to that credit consumption prompt, the FBAC will grant the request q_i by charging the requester corresponding fee, i.e., deducting $cost(q_i)$ from c_x . In fact, for individuals, the FBAC would degrade to standard ABAC when they max out their credits.

Furthermore, for achieving better credit management and thereby controlling credit abuse risks, a periodic manual audit mechanism is also integrated into the FBAC model. During an audit, the unusual authorization records will be reviewed by the system administrators according to all the relevant information in the system including corresponding

explanatory comments typed by requesters in the confirmation process. Based on auditing results, the audit routine will restore credits for the users who pass checks successfully, while disables such recovery for the suspects unless proved innocent (More tougher punishments can be given when the suspect is finally proven guilty.), to ensure the credit system works well, thereby providing enough flexibility with controllable abuse risks.

Note that the credit recovery strategy depends on the administrator. For instance, our approach gives the proportional credit back (r in 100%) of the margin between the credit line c_{max} and the current credit value c_x (i.e., $c_{max} - c_x$) after each audit process. This is because we hold a conservative opinion that the special approval is a compromise for improving business timeliness, which should not be encouraged in routine work. Therefore, the formula for calculating new credit value c'_x is as follows:

$$c'_x = r(c_{max} - c_x) + c_x, \quad \text{where } r \in (0, 100\%]. \quad (4)$$

4.2. Case Study. This subsection provides a case study of FBAC to help people understand how it works in detail.

Assuming there exists an FBAC system with the threshold $H = 0.8$, $c_{max} = 0.3$, $r = 0.5$ and two clauses in the policy set as follows:

$$\text{policy} : \begin{cases} (1) \text{ IF } (location = (112.54153E \pm 0.00001, 28.95117N \pm 0.00001)) \\ \quad \text{and } (job \text{ title is manager}) \text{ THEN granted} \\ (2) \text{ IF } (location = (112.54153E \pm 0.00001, 28.95117N \pm 0.00001)) \\ \quad \text{and } (time \in [8 : 00, 18 : 00]) \text{ and } (job \text{ title is staff}) \text{ THEN granted} \end{cases} \quad (5)$$

We can see that there are 3 types of attributes involved in the policy set: *time* is the timestamp of the request, *location* denotes the requester's location (given in latitude and longitude), and *job title* denotes the *subject's* job position. Then we define the membership functions as follows:

$$\begin{aligned} \mu(q_i) &= \max(v_1(q_i), v_2(q_i)) \\ v_1(q_i) &= \frac{\sum_{j=1}^2 w_{1,j} \xi_{1,j}(q_i)}{\sum_{j=1}^2 w_{1,j}} \\ v_2(q_i) &= \frac{\sum_{j=1}^3 w_{2,j} \xi_{2,j}(q_i)}{\sum_{j=1}^3 w_{2,j}} \end{aligned} \quad (6)$$

In this case, we set all the attributes in the same policy to the same weight, as shown below:

$$\begin{aligned} v_1(q_i) &= \frac{\sum_{j=1}^2 \xi_{1,j}(q_i)}{2} \\ v_2(q_i) &= \frac{\sum_{j=1}^3 \xi_{2,j}(q_i)}{3} \end{aligned} \quad (7)$$

In order to describe $\xi_{i,j}$, we firstly predefine a function $distance(x, y)$ to describe the distance between x and y in meters. Then, we give the definitions of $\xi_{i,j}$ as follows:

$$\begin{aligned} \xi_{1,1}(q_i) &= \max\left(1 - \frac{distance(location, office)}{100}, 0\right) \\ \xi_{1,2}(q_i) &= \begin{cases} 1 & \text{job title is manager} \\ 0 & \text{otherwise} \end{cases} \\ \xi_{2,1}(q_i) &= \xi_{1,1}(q_i) \\ \xi_{2,2}(q_i) &= \begin{cases} 2 \cdot time - 16, & time \in (7.5, 8] \\ 1, & time \in (8, 18] \\ 37 - 2 \cdot time, & time \in (18, 18.5] \\ 0, & \text{otherwise} \end{cases} \\ \xi_{2,3}(q_i) &= \begin{cases} 1, & \text{job title is staff} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (8)$$

Then we assume that a subject S initiates a request q_1 as follows:

$$q_1 = \left\{ \begin{array}{l} \text{time} = 18 : 35 \\ \text{job title} = \text{manager} \\ \text{location} = (112.54180E, 28.95117N) \end{array} \right\} \quad (9)$$

When request q_1 is initiated, the FBAC attempts to match q_1 with policies but fails. Then it turns to the fuzzy evaluation process. As the credit cost of the q_1 is $\text{cost}(q_1) \approx 1 - 0.85 = 0.15$, then 0.15 is going to be consumed from c_s for making q_1 be granted. The system will ask subject S for the consumption confirmation in order to make sure whether S is willing to consume required credits to continue. Suppose that S chooses to spend his credits, then q_1 is granted, and c_s is decreased to 0.15.

Next, when S try to initiate another request q_2 later as follows:

$$q_2 = \left\{ \begin{array}{l} \text{time} = 23 : 03 \\ \text{job title} = \text{manager} \\ \text{location} = (112.54187E, 28.95117N) \end{array} \right\}, \quad (10)$$

in the same way, we get that $\text{cost}(q_2) \approx 0.19$. Since $c_s = 0.15$ after the request q_1 , S can not afford the cost of the q_2 , so q_2 will be rejected directly.

In addition, if S passes the audit with his credit value $c_s = 0.15$, then c_s will be restored to 0.225 according to expression (4).

5. Discussion

In this section, we will briefly analyze the effect on usability and security of FBAC, followed by complexity analyses.

Usability and Security. To describe the enhance effect on the overall resource usability of FBAC, we chose the granted rate, which is defined as the rate of the granted requests to total requests per unit time, as a reflection of usability.

Let U denote the usability and R denote the granted rate; then we get the following expression in which R_{normal} and $R_{special}$ denote the granted rates of requests matching or not matching policies, respectively, while notation “ \propto ” denotes the relationship of positive correlation.

$$U \propto (R = R_{normal} + R_{special}) \quad (11)$$

Since FBAC shares the same R_{normal} with its elder sibling ABAC obviously, the FBAC obtains extra usability improvement ΔU which is positively correlated with $R_{special}$ when compared with ordinary ABAC, namely,

$$\Delta U \propto R_{special} \quad (12)$$

Naturally, the configurable threshold H is closely associated with the usability. For any request q_* failed in policies matching with overall matching degree $\mu(q_*)$, we suppose that $\mu(q_*) = x$ obeys a probability density distribution $f(x)$

while the probability of available credit of requester $c_* \geq \mu(q_*)$ obeys another probability density distribution $h(x)$, then we can deduce the following relational expression:

$$R_{special} \propto \int_H^{Max} h(x) f(x) dx \quad (13)$$

Since $h(x)$ and $f(x)$ are commonsensically positive, we find an inverse correlation between the incremental usability ΔU and the threshold H in expression (13); that is, a lower H leads to more approvals on requests. Apparently, the FBAC would deteriorate to standard ABAC if H tends to the upper bound, i.e., the value 1 in our case.

Not surprisingly, the usability improvement also comes with security risks. As the FBAC may authorize exceptional access requests which do not fully comply with the current policies in some cases, this feature can be abused by indiscreet users or even be exploited by malicious users for accessing extra resources and thereby bringing additional risks to the system. Here, the deviation between the overall matching degree of the exceptional request (i.e., $\mu(q_*)$) and the closest matching policy (the standard normalization value “1”) is used as the risk indicator of each exceptional authorization.

Correspondingly, the FBAC has effective countermeasures to mitigate the risks induced by the fuzzy assessment mechanism to the acceptable level. Firstly, as a general and indiscriminate defense, the reject threshold is used to screen out high-risk requests deviating far from current policies, i.e., any request q_* with overall matching degree $\mu(q_*)$ lower than the threshold H would be declined directly, because the FBAC is aiming at improving the flexibility and efficiency of exceptional authorizations rather than invalids the security policies. Thus, the security risk of each exceptional authorization is limited within the controllable range $1 - H$. Secondly, the credit mechanism is used as the individualized constraint against the abuse attacks on the FBAC. As for each requester, each exceptional authorization definitely comes with corresponding credit cost which is determined by the risk of that request q_* (i.e., $\text{cost}(q_*) = 1 - \mu(q_*)$). In other words, a request q_* will be declined if the corresponding requester x_* does not have enough credit to afford the credit cost $\text{cost}(q_*)$ of the exceptional request, i.e., $c_{x_*} < \text{cost}(q_*)$. Therefore, the immoderate and even malicious exceptional access behaviors are mitigated due to the limitation of credit. According to the analysis above, then the maximum security risk of one exceptional authorization associated with a requester x_* is further limited within $\text{Minimum}(1 - H, c_{x_*})$. Meanwhile, within each audit cycle, the total security risk which can be caused by the exceptional authorizations related to each single requester x_* is limited below his credit value c_{x_*} (the value at the beginning of the audit cycle). In addition, for each subject x_* , the credit consumption has the additive restrictive effect on future requests because only a portion of the already consumed credits could be restored according to credit recovery mechanism. Briefly, the more credits the requester used in one audit cycle, the less total amount he will have in the future, which further reduces the abuse risks of the exceptional authorizations. Finally, the FBAC integrates a periodic manual audit mechanism as

TABLE 2: The parameter configuration.

| Case | C_{max} | r | H | Time weight | Location weight |
|------|-----------|------|------|-------------|-----------------|
| 1 | 0.80 | 0.50 | 0.80 | 0.50 | 0.50 |
| 2 | 0.80 | 0.50 | 0.85 | 0.50 | 0.50 |
| 3 | 0.80 | 0.50 | 0.90 | 0.50 | 0.50 |
| 4 | 0.80 | 0.50 | 0.80 | 0.40 | 0.60 |
| 5 | 0.80 | 0.50 | 0.80 | 0.20 | 0.80 |

the post-security mechanism to review all the exceptional authorizations. As for the suspects, their credit restorations would be suspended until proven innocent. As a result, they would lose the privileges to obtain instant approvals for their exceptional requests as their credits will keep reducing and can not get replenished. Therefore, the entire risk which can be caused by the exceptional authorizations granted for a single suspect identified during the audits is limited within the credit line c_{max} .

Summarily, the FBAC broadens the granting bounds to a certain extent for all the requests with the help of fuzzy evaluation mechanism and limits the special approval rate of each individual requester with the help of credit and audit mechanism, thereby achieving better timely usability than standard ABAC with the controllable sacrifice of security.

Complexity. The complexity of access control is related to the number of concurrent requests, policies, and attributes contained in each policy. The more the attributes are involved in a policy, the higher the computational complexity of this policy will be. Generally, as the granularity of access control becomes finer, the complexity of policy increases and the time cost of decision-making process also grows slightly and tends to flatten out.

Assuming there are m policies and n attributes, the number of requests that occur at the same time in the system is k , the computational complexity of a basic matching process is $O(1)$ in original ABAC model. In the worst case, each policy and attribute needs a matching calculation, and thus the complexity of a single decision is $O(mn)$. Because complexity is proportional to the number of requests made

simultaneously, the total computational complexity of the whole system is $O(kmn)$.

Correspondingly, the computational complexity of both a basic matching process and credit evaluation process in our FBAC model is also $O(1)$; that is to say, the complexity of a single decision is still $O(mn)$; thus the total computational complexity remains at $O(kmn)$.

Compared with the standard ABAC model, our FBAC model has two additional processes, the credit-based judgment and the fuzzy assessment, which is a little complex than the simple yes/no decision. And the overhead of both parts can be considered of the same order of magnitude as the former. This explains why both models (i.e., ABAC and FBAC) have the same computational complexity. It also shows that the impact of FBAC in terms of performance is within an acceptable range.

6. Experimental Evaluation

We developed an FBAC prototype to evaluate its availability, security, and performance through several experiments.

6.1. Test Scenarios. By modifying the ABAC source codes of Deter Project [35], we implemented a prototype of FBAC and deployed it to 5 virtual servers on a single physical machine (64-bit CentOS 7, 4vCPUs (i5-7500 3.4GHz), 16GB RAM, 1TB Storage, supported by OpenStack (Pike v3.12.0)) for experiments.

In our FBAC systems, we firstly configured the following policy set and set the audit time interval to one week uniformly.

$$policy : \begin{cases} \text{IF } (location = (112.54153E \pm 0.0001, 28.95117N \pm 0.0001)) \\ \text{and } (time \in [8 : 00, 18 : 00]) \text{ THEN granted} \end{cases} \quad (14)$$

And then we conducted four experiments with respective FBAC configuration parameters shown in Table 2. And in each experiment, we simulated 500 users to initiate requests to FBAC servers. These users follows Poisson distribution in time and move around according to Random Way Point (RWP) [36] model to fit the mobile features. The simulation system will randomly regenerate the destination and the moving speed for each user every 30 minutes. Additionally, we also introduced small noises ($\pm 10m$) randomly to

users' location coordinate data for simulating the fluctuations in the real positioning system. These users were set as "benign" or "malicious" separately with several different user behavioral patterns correspondingly to generate requesting data. Furthermore, we set that benign users will abort their requests randomly in responding to credit misuse prompts whereas malicious users will not, according to the knowledge that benign users are more compliance with rules.

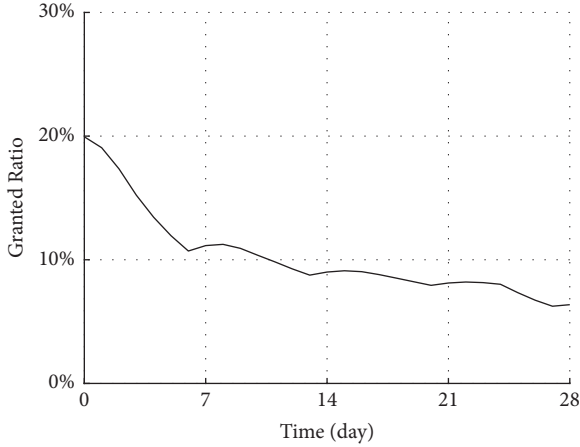
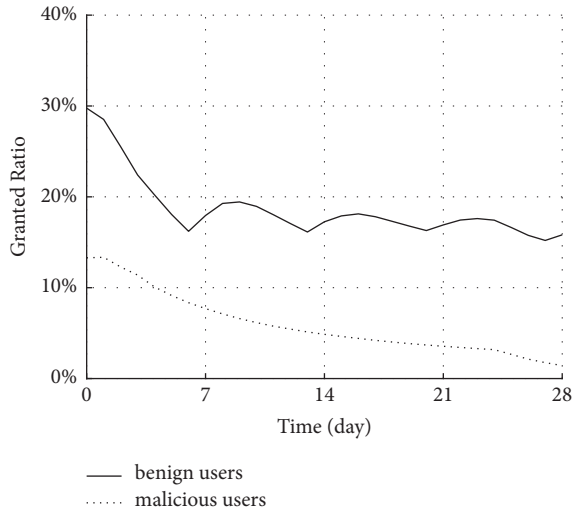


FIGURE 2: The average granted ratio of requests.

FIGURE 3: $R_{special}$ of benign and malicious users.

Note that in the fourth and fifth cases, we forced all the users to obey the time restriction to articulate the effect of attribute weights.

The experiments last for four weeks and each audit period is 5 days long. All the access histories are recorded in access logs for further analyses.

6.2. Analysis

Usability. As the granted ratio of requests which fail to meet policies (denoted by $R_{special}$) reflects the extra improvement on immediate resource usability, we count up such average granted rate based on the Case 1, as shown in Figure 2. We can learn that the average granting rate of exceptional requests is maintained in a positive range during the experiment, which illustrates the usability increment of FBAC compared with ABAC through the employment of fuzzy evaluation method.

Security. Again, based on Case 1, we evaluated the resistance of FBAC against security risks. Figure 3 shows the granted

TABLE 3: The time cost of the decision-making process.

| Model | Average time (ms) | Best time (ms) | Worse time (ms) |
|-------|-------------------|----------------|-----------------|
| FBAC | 0.033 | 0.019 | 0.245 |
| ABAC | 0.017 | 0.002 | 0.081 |

ratios of both benign and malicious user respectively. It is clear that $R_{special}$ of benign users is limited to a certain upper bound by the threshold, particularly, below 35% in Case 1, while that of malicious users is even far lower throughout the test duration. Furthermore, it also illustrates that such rates of both benign and malicious users are further constrained by credit mechanism. With the consumption and partial recovery of credits controlled by credit and audit mechanism, $R_{special}$ of benign users reveals a hysteretic declined trend within each audit cycle and will fluctuate along with audit cycles during the testing period. When it comes to malicious users, this ratio is decreasing continuously over audit cycles and is gradually converging to 0.

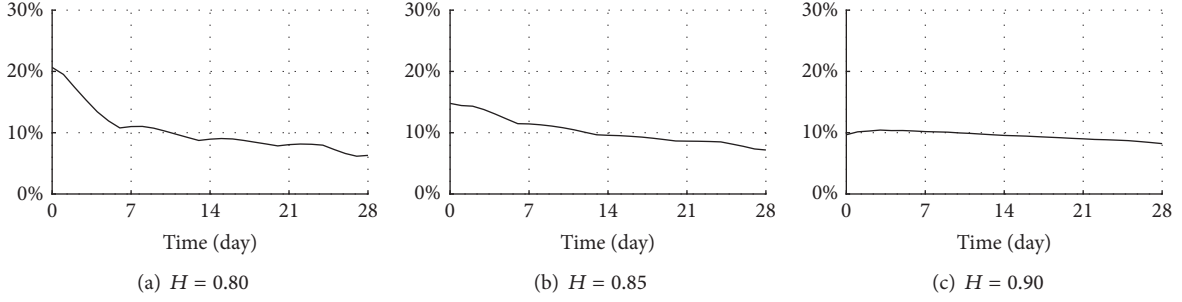
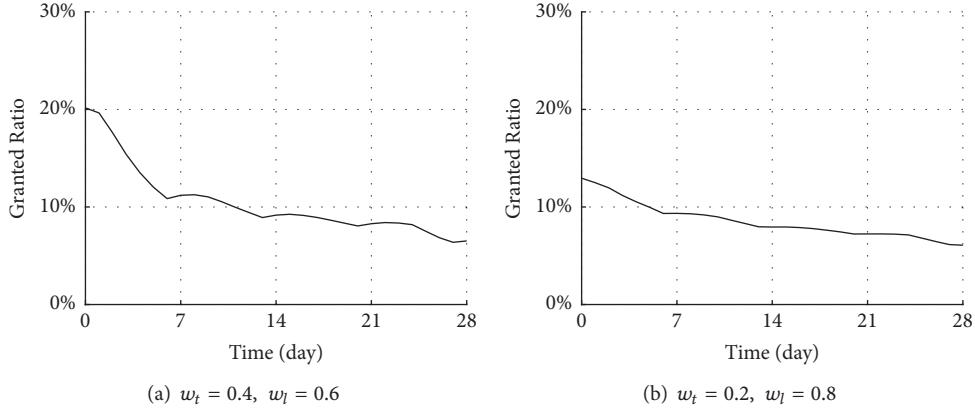
Such results demonstrate that the threshold provides a general and coarse-grained restriction on requests while credit system supplies additive restrictive effect on the requests in each audit cycle. In addition, the audit mechanism is effective in limiting $R_{special}$ of users with malicious or abnormal behaviors as their credits will be used up easily and can hardly be restored because of the audit mechanism. Therefore, the FBAC is sufficient to defend against abuse attacks.

Parameter Effects. We have tuned two major regulative parameters in FBAC to explore their potential influence.

(1) **Threshold.** To study the impact of the reject threshold, we increased the threshold H by 0.05 in Case 1, Case 2 and Case 3 gradually. Unsurprisingly, Figure 4 illustrates that $R_{special}$ in FBAC is closely related to the threshold H ; i.e., the higher H is, the lower the granted rate will be. Besides, although a low H may accelerate the credit consumption, which in turn affects the granted rate due to the rejection cases caused by credit insufficiency, this side effect is unable to impact the main trend on a macroscale.

(2) **Attribute Weight.** When it comes to the attribute weight, Cases 4 and 5 were selected for comparison as they set the time variable to fixed value by obeying the time restriction and share the same C_{max} and H parameters. As seen in Figure 5, the bigger weight coefficient for the location attribute in Case 5 leads to a lower granted rate when compared with that of Case 4. This shows that the weight mechanism can effectively adjust the overall impact of each attribute on the decision-making process.

Performance. We evaluated the time cost of decision-making processes of both FBAC and ABAC to measure the performance. According to the results in Table 3, although FBAC wraps ABAC and adds additional mechanisms for making authorization decisions, it only incurs quite light and

FIGURE 4: $R_{special}$ under different thresholds.FIGURE 5: $R_{special}$ under different attribute weights.

acceptable overhead in average compared with ABAC, which is almost imperceptible to requesters.

7. Conclusion

In this paper, a feasible FBAC technique is proposed that improves upon the standard ABAC paradigm with good flexibility and time efficiency in dealing with exceptional urgent requests which do not comfort to policies in the dynamic and unpredictable environment. Beyond ABAC, we use a fuzzy evaluation method to do unattended special authorizations for exceptional requests that failed in policy matching. We also use credit and corresponding audit mechanisms to limit the abuse risk of special approvals. A tangible example is given to explain the working details, which indicates the suitability of FBAC in mobile and dynamic scenarios. In addition, the theoretical analyses and experimental evaluations show that the FBAC paradigm reinforces the system in favor of time efficiency and usability with the controllable expense of security.

In future work, we would like to further refine the authorization decision-making scheme with the support of the latest deep learning techniques (e.g., neural network) to discover benign and riskful access patterns based on the access behavior mining for helping the FBAC better distinguish between benign and malicious requests, thereby

enabling more intelligent and accurate handling for exceptional access cases. Moreover, we also believe that deploying the FBAC system in China's current Xiangya medical big data system would have more practical and exploratory meanings.

Disclosure

This work was presented in part at the SpaCCS 2017, Guangzhou, China, 12–15 December 2017.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61702562 and 61472451, the Mobile Health Ministry of Education-China Mobile Joint Laboratory, the Hunan Provincial Innovation Foundation for Postgraduate under Grant CX2015B047, the China Scholarship Council Foundation under Grant 201506370106, the Guangdong Provincial Natural Science Foundation under Grant 2017A030308006, and the Joint Research Project between Tencent and Central South University.

References

- [1] G. Fettweis and S. Alamouti, "5G: personal mobile internet beyond what cellular did to telephony," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 140–145, 2014.
- [2] Y. Zhang, K. Guo, J. Ren, J. Wang, and J. Chen, "Transparent computing: A promising network computing paradigm," *Computing in Science Engineering*, vol. 19, no. 1, p. 20, 2017.
- [3] Y. Zhang, J. Ren, J. Liu, C. Xu, H. Guo, and Y. Liu, "A survey on emerging computing paradigms for big data," *Journal of Electronics*, vol. 26, no. 1, pp. 1–12, 2017.
- [4] J. He, Y. Zhang, J. Lu, M. Wu, and F. Huang, "Block-Stream as a Service: A More Secure, Nimble, and Dynamically Balanced Cloud Service Model for Ambient Computing," *IEEE Network*, vol. 32, no. 1, pp. 126–132, 2018.
- [5] T. Peng, Q. Liu, and G. Wang, "A multilevel access control scheme for data security in transparent computing," *Computing in Science & Engineering*, vol. 19, no. 1, Article ID 7802524, pp. 46–53, 2017.
- [6] I. Hardill and A. Green, "Remote working - Altering the spatial contours of work and home in the new economy," *New Technology, Work and Employment*, vol. 18, no. 3, pp. 212–222, 2003.
- [7] A. M. French, C. Guo, and J. P. Shim, "Current status, issues, and future of bring your own device (BYOD)," *CAIS*, vol. 35, pp. 1–10, 2014.
- [8] D. F. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.
- [9] S. Upadhyaya, "Mandatory access control," in *Encyclopedia of Cryptography and Security*, pp. 756–758, Springer, 2011.
- [10] L. Liu and M. Tamer Özsu, "Discretionary access control," in *Encyclopedia of Database Systems*, pp. 864–866, Springer, 2009.
- [11] V. C. Hu, D. Ferraiolo, R. Kuhn et al., "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," National Institute of Standards and Technology NIST SP 800-162, 2014.
- [12] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-based access control," *The Computer Journal*, vol. 48, no. 2, Article ID 7042715, pp. 85–88, 2015.
- [13] D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control," *ACM Computing Surveys*, vol. 49, no. 4, article no. 65, 2017.
- [14] X. Li, D. Feng, Z. Chen, and Z. Fang, "Model for attribute based access control," *Journal on Communications*, vol. 29, no. 4, pp. 90–98, 2008.
- [15] X. Jin, *Attribute-based access control models and implementation in cloud infrastructure as a service [Ph.D. thesis]*, The University of Texas at San Antonio, 2014, Ph.D. dissertation.
- [16] M. Sookhak, F. R. Yu, M. K. Khan, Y. Xiang, and R. Buyya, "Attribute-based data access control in mobile cloud computing: Taxonomy and open issues," *Future Generation Computer Systems*, vol. 72, pp. 273–287, 2017.
- [17] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman, "A flexible attribute based access control method for grid computing," *Journal of Grid Computing*, vol. 7, no. 2, pp. 169–180, 2009.
- [18] C. Ngo, Y. Demchenko, and C. De Laat, "Multi-tenant attribute-based access control for cloud infrastructure services," *Journal of Information Security and Applications*, vol. 27–28, pp. 65–84, 2016.
- [19] Y. Xu, W. Gao, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "FABAC: A flexible fuzzy attribute-based access control mechanism," in *Proceedings of the Proc. 10th International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 332–343, Springer, 2017, pp. 332–343.
- [20] X. Liu, Q. Liu, T. Peng, and J. Wu, "Dynamic access policy in cloud-based personal health record (PHR) systems," *Information Sciences*, vol. 379, pp. 62–81, 2017.
- [21] IBM Corporation, https://www.ibm.com/support/knowledgecenter/en/SSNGTE_7.0.0/com.ibm.tspm.doc_7.0/install/concept/AttributeBasedAccessControl.htm.
- [22] "Cisco Systems, Inc," <https://www.cisco.com/c/en/us/td/docs/security/asa/asa97/asdm77/firewall/asdm-77-firewall-config/virtual-access-vm-attributes.pdf>.
- [23] Axiomatics, <https://www.axiomatics.com/>.
- [24] Jericho Systems Corporation, https://www.jerichosystems.com/technology/glossaryterms/attribute_based_access_control.html.
- [25] C. Martnez-Garca, G. Navarro-Arribas, and J. Borrell, *Fuzzy role-based access control*, vol. 111 of *Information Processing Letters*, Elsevier, 2011, pp. 483–487.
- [26] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, "Fuzzy multi-level security: an experiment on quantified risk-adaptive access control," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 222–230, IEEE, Berkeley, Calif, USA, May 2007.
- [27] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, and K. Moody, "Using trust and risk in role-based access control policies," in *Proceedings of the Proceedings on the Ninth ACM Symposium on Access Control Models and Technologies, SACMAT 2004*, pp. 156–162, usa, June 2004.
- [28] Y. Li, *The research of access control mechanism based on attribute and trust evaluation*, Masters thesis [Master, thesis], Southwest Jiaotong University, 2016.
- [29] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad, "A fuzzy approach to trust based access control in internet of things," in *Proceedings of the Proc. 3rd International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)*, pp. 1–5, 2013.
- [30] F. Feng, C. Lin, D. Peng, and J. Li, "A trust and context based access control model for distributed systems," in *Proceedings of the Proc. 10th IEEE International Conference on High Performance Computing and Communications*, pp. 629–634, 2008.
- [31] R. Bhatti, E. Bertino, and A. Ghafoor, "A trust-based context-aware access control model for web-services," *Distributed and Parallel Databases*, vol. 18, no. 1, pp. 83–105, 2005.
- [32] F. J. Pelletier, "Metamathematics of fuzzy logics by Petr Hajek," *Bulletin of Symbolic Logic*, vol. 6, no. 3, pp. 342–346, 2000.
- [33] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.
- [34] J. Dombi, "Membership function as an evaluation," *Fuzzy Sets and Systems*, vol. 35, no. 1, pp. 1–21, 1990.
- [35] DETER Project, <https://abac.deterlab.net/>.
- [36] D. Johnson and D. Maltz, "Dynamic source routing in Ad Hoc wireless networks," in *The Kluwer International Series in Engineering and Computer Science*, vol. 353, pp. 153–181, 1996.

Research Article

Demadroid: Object Reference Graph-Based Malware Detection in Android

Huanran Wang , Hui He , and Weizhe Zhang 

Department of Computer Science and Technology, Harbin Institute of Technology, 92 Xidazhi Street, Harbin, Heilongjiang 150001, China

Correspondence should be addressed to Hui He; hehui@hit.edu.cn and Weizhe Zhang; wzzhang@hit.edu.cn

Received 29 December 2017; Accepted 11 April 2018; Published 31 May 2018

Academic Editor: Dafang Zhang

Copyright © 2018 Huanran Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Smartphone usage has been continuously increasing in recent years. In addition, Android devices are widely used in our daily life, becoming the most attractive target for hackers. Therefore, malware analysis of Android platform is in urgent demand. Static analysis and dynamic analysis methods are two classical approaches. However, they also have some drawbacks. Motivated by this, we present Demadroid, a framework to implement the detection of Android malware. We obtain the dynamic information to build Object Reference Graph and propose λ -VF2 algorithm for graph matching. Extensive experiments show that Demadroid can efficiently identify the malicious features of malware. Furthermore, the system can effectively resist obfuscated attacks and the variants of known malware to meet the demand for actual use.

1. Introduction

Android is a mobile operating system developed by Google, based on the Linux kernel, and designed primarily for touchscreen mobile devices such as smartphones and tablets [1]. On top of the kernel level, there are middleware, libraries, and APIs written in C programming language. And the kernel level is independent of other resources [2].

With the popularity of smartphones, the number of users of Android dramatically rises [3]. However, the popularity of Android also attracts the attention of malware, which has become an urgent threat to users [4]. According to the types of threats, malicious apps can be divided into at least six categories: abuse of value-added services software, advertising fraud software, data theft software, malicious downloading software, malicious decoding software, and spyware. Research from security company Trend Micro shows that the premium service abuse is the most common type. For example, text messages are sent from infected phones without the permission of users [5]. Android has become the hardest hit. However, Google engineers have argued that the malware and virus threat on Android is being exaggerated by security companies for commercial reasons. A survey published by F-Secure showed that only 0.5% of Android malware reported had come from the Google Play store [6].

In addition, the source of malware is very extensive. Different from the PC virus, Android malicious attack has its own features; various types of malicious codes cover almost every level. The proportion of various malware types is shown in Figure 1 [7].

Motivated by this, a great number of Android malware detecting methods are proposed which are divided into two types as follows [8].

The first kind of methods is static analysis. Static methods analyze the executable file directly instead of running it. For example, DroidDet [9] statically detects malware by utilizing the rotation forest model. However, this work cannot resist the obfuscated attack.

Another type of approaches is dynamic analysis. Different from the static methods, dynamic methods extract the malicious features at runtime, which improves the effectiveness of detection. By contrast, dynamic analysis has stronger robustness. Dynamic analysis techniques are not compatible in some cases because developing tools that allow the dynamic analysis of malware is very challenging, and such techniques require extensive resources and often do not have enough scale to be used in practice [10]. Shabtai A et al. [11] propose a new dynamic technique, sandbox, which is built by the kernel LKM (Loadable Kernel Module). They analyze the

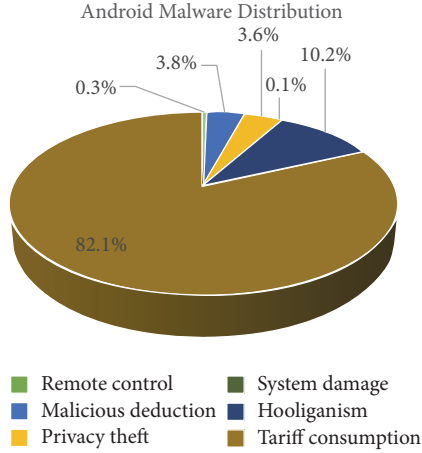


FIGURE 1: Android malware distribution. This figure is reproduced from 360 Internet Security Center [7] (2017) [under the Creative Commons Attribution License/public domain].

system calls from the kernel to create the log file. However, the modification to the kernel level causes the instability of operations, and the user interaction is only simulated by automatic tools, which is no real operation [12].

To address these problems, we propose a more effective Android dynamic technology to detect malware. This is a new technique of establishing dynamic birthmarks. We extract the reference relationships between objects allocated in heap memory and then establish ORG (Object Reference Graph) to build ORGB (Object Reference Graph Birthmark) as the feature. In addition, we propose λ -VF2 algorithm to match the subgraph isomorphism.

Compared with the existing dynamic birthmark methods, we utilize the information in heap, which can also be used to solve the problem of code plagiarism. In summary, the main contributions of this paper are listed as follows.

- (i) We establish ORG by extracting all the referential relationships between objects allocated in heap memory.
- (ii) With the analysis of the program class, we extract the feature classes to build ORGB as the birthmark of malware.
- (iii) Based on VF2 algorithm, we propose λ -VF2 algorithm to improve the false negative rate and false positive rate.
- (iv) We propose an Android malware detection system Demadroid which resists the obfuscated attack. To demonstrate the effectiveness of the proposed approaches, we conduct extensive experiments. Experimental results show that the proposed system and algorithm perform well.

The rest of the paper is organized as follows. In Section 2, we discuss the related work, and we give the details of our algorithm in Section 3. Section 4 presents the framework of Demadroid. The evaluation of Demadroid is depicted in Section 5. In Section 6, we summarize the whole work.

2. Related Work

Several approaches have been proposed recently to detect malware in Android. Generally, they are divided into static analysis and dynamic analysis.

Static analysis inspects app without executing it. Julia is a Java bytecode static tool for Android platform, but it cannot parse the classes generated by the XML file mapping. Payet É et al. [10] improve it to analyze the bytecode of Dalvik Virtual Machine. Kui Luo et al. [13] propose a bytecode conversion tool for privacy stolen malware and enable it to convert into DVM bytecodes and analyze Android programs. Literature [14] uses the existing tools dex2jar and FindBugs for analysis, which traversed the flowchart of Android programs and obtains the functional dependencies between Intent objects. The above works are based on existing tools, which have a great number of limitations. Batyuk L et al. [15] present disassembly method by disassembling the malicious code of Android. They get the malicious part and modify it to separate the malicious code. This method is effective for the untreated apps but cannot deal with the obfuscated code. Based on sensitive data access, Di Cerbo F et al. [16] study the privacy-stealing code. By analyzing the permissions feature of the program request, they compare with the defined features to determine whether the program is malicious. One important problem in this work is that Android does not have permission restrictions on the use of API. Therefore, it cannot identify the malicious code utilizing Android vulnerabilities. In a word, the drawbacks of static methods are obvious; their robustness is weak. And several attacks such as code obfuscation, Junk Code, and other antidetection techniques can easily avoid detection.

Dynamic analysis can resist the code obfuscation attack but is more expensive than static methods. Isohara T et al. [17] use a kernel-level monitoring method to record the system call of Android program. This method can effectively analyze the record of system calls. However, it is just used for the monitoring of stolen information. Based on this, Schmidt A D et al. [18] present further research and divide the monitoring into Android application layer, system application layer, and system kernel layer. However, there are no valid experimental tests to verify the feasibility of the work. Crowdroid [19] is a classifier based on anomaly detection. The system uses the existing Strace program to monitor system calls and create record files. After being uploaded to the server, the files are classified by the K-Means algorithm. However, in this case, the amount of data and the network traffic of the system are relatively large, and the problem of data security is brought at the same time. Attackers can easily fabricate the key information and interfere with the result. Shabtai A et al. [11] mention a dynamic analysis technique, sandboxing, which is a new direction for Android malicious code detection. However, the current sandbox technology is incomplete. Myles et al. [20] use the control flow of apps to identify malicious behaviors. Experiments show that control-flow analysis is more effective than static birthmark analysis in dealing with attacks utilizing the semantics.

3. VF2 Algorithm

3.1. Isomorphic Patterns of Graphs. In the past decades, graph matching has been one of the main research topics in computer science. In general, graph matching can be classified into two lines, exact-matching algorithms and inexact-matching algorithms. Exact-matching algorithms require strict consistency between two candidate graphs. The most stringent pattern of exact-matching algorithms is graph isomorphism, which requires the mapping of nodes and edges on both graphs to be bijections [21]. The fuzzier pattern of exact-matching is subgraph isomorphism which requires at least the strict consistency between the subgraph and the ideograph [22].

Moreover, inexact-matching algorithms, which are also called fault-tolerance matching, relax the constraints with errors and noises. Monomorphism is the inexact-matching which gets rid of the bidirectional requirement of edge-remaining bases on subgraph isomorphism. It requires that every node of the first graph can map different nodes and edges in the second graph, which allows the redundant edges and nodes. The weaker graph match pattern is the homomorphism, which is a many-to-one mapping that does not require that every node of the first graph is mapped to a different node of the second graph. Isomorphism matching is another method to match the subgraphs, of which the result is not unique. It is also used to find the largest subgraph match, which is called the maximum common subgraph (MCS).

3.2. Analysis of the Subgraph Isomorphism Matching Algorithm. All the isomorphic patterns are NP-complete problems except graph isomorphism. Whether graph isomorphism is NP-complete problem has not been proved till now [23]. At present, polynomial time algorithms are matched for special types of graphs, and there is no general polynomial time algorithm for general graphs. For this reason, the time complexity of the exactly matching algorithm is exponential in the worst case. However, in practical problems, the cost of time is basically acceptable. Because the type of graph encountered in practical problems is not the worst case and the attributes of the nodes and edges can greatly reduce the search time.

The problem of graph isomorphic matching is a very classic problem in graph theory, and the algorithms used in different scenarios are different. In practice, the data required for the establishment of a graph will inevitably be disturbed; that is why graph isomorphism is rarely used. Subgraph isomorphism and monomorphism are commonly used patterns. They are more effective in dealing with practical problems. Many algorithms have been developed for these two problems. At present, the exact match algorithm is more effective for the basic graphs and searching for MCS.

3.2.1. Ullmann Algorithm. One of the most important types of graph matching algorithm is the Ullmann algorithm [24], which was proposed in 1976. It can solve the isomorphic problems, such as isomorphism, subgraph isomorphism,

and monomorphism. At the same time, the algorithm also provides a way to deal with the maximum matching, so it can also be used to solve the CMS problem.

To reduce the bad matching branches, Ullmann algorithm proposes predictive equation to control backtracking process, significantly reduce the scale of search space, and improve the performance of the algorithm.

3.2.2. Ghahraman Algorithm. Ghahraman proposed another backtracking based monomorphism algorithm in 1980 [25]. To reduce the search space, a technique like association graph is used in this paper. The matching search is carried out on the NetGraph matrix. This matrix is generated by the product of the Descartes product between the nodes of the matched two graphs. The monomorphism matching of the two graphs is related to a subgraph of the NetGraph. The author finds two necessary conditions for the partial matching to produce the result.

One of the main disadvantages is that the storage of NetGraph requires at least one matrix of $N_2 * N_2$ size, in which N represents the number of nodes. Therefore, this algorithm is more suitable for a graph with lower number of nodes.

3.2.3. Nauty Algorithm. Nauty algorithm [26] is the most famous tree search algorithm which is not based on backtracking. It only deals with the isomorphic problem and is recognized as the fastest one. By using the conclusion group theory, it creates an automorphism group for each input. And every automorphism group produces a standard label to guarantee that the only node order is introduced by each equivalent class of the automorphism group. Then, the isomorphic comparison of the two graphs is equivalent to the adjacency matrix comparison of the standard label.

The time complexity of comparison is $O(N_2)$ of the worst case. In most cases, the time performance is acceptable. Because the establishment of standard tags can be carried out independently. Therefore, it is more suitable for the graph matching in a large library.

3.2.4. VF and VF2 Algorithm. The VF algorithm proposed by Cordella [27] is applied to both isomorphism and subgraph isomorphism. Cordella defined a heuristic search by analyzing the adjacent nodes of matched nodes. This heuristic algorithm is significantly better than Ullman and other algorithms in many cases.

Cordella improved the algorithm in 2001, which is called the VF2 algorithm [28]. The improvement reduces the space complexity from $O(N_2)$ to $O(N)$, in which N donates the number of nodes. In this way, the algorithm can be applied to the matching of large graphs.

The VF2 algorithm is also used in many other related fields. For example, Jonathan Crussell et al. propose DNADroid [29], a tool which uses VF2 algorithm to detect cloned apps. In this work, VF2 algorithm is used to compute subgraph isomorphism. The experiment proves that VF2 algorithm is suitable for graphs containing a variety of node types.

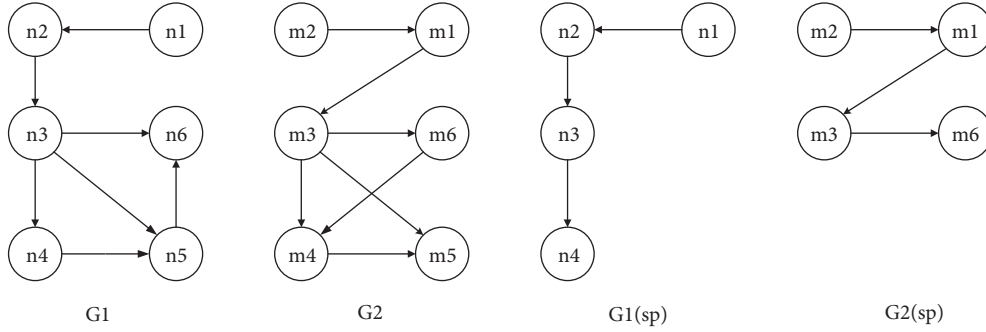


FIGURE 2: SSR instance diagram.

3.3. Comparison of Subgraph Isomorphism Matching Algorithms. In this section, we analyze several classical algorithms mentioned in Section 3.2 and select the proper algorithm as the foundation of our matching process. The main types of graph include the bounded Valence Graph, the two-dimensional grid graph (2D Mesh Graph), and the random connection graph (Randomly Connected Graph). Foggia et al. analyze the above algorithms by experiments [30]. The ORG in our work is similar to random connection graph and quite different from the other two kinds. Therefore, we only discuss the condition of random connection graphs. Foggia uses a control group with different density of nodes and edges. The experimental result shows that VF2 algorithm and Nauty algorithm are better than Ullmann algorithm in dealing with random connection graphs. VF2 performs better than VF algorithm when the density is different. Compared with Nauty algorithm, VF2 algorithm has a better effect to match sparse graphs. And Nauty algorithm is more applicable to dense graphs.

In this paper, we match the subgraphs between object reference dependency graphs, in which nodes represent classes, and directed edges represent references between classes. According to the analysis of samples, the number of nodes in ORG is within 100. Therefore, the algorithm used in this paper is based on VF2 algorithm.

3.4. Review of VF2 Algorithm. VF2 algorithm is applicable to isomorphism, subgraph isomorphism, and monomorphism because it does not impose restrictions on the topology of matched graphs. The algorithm adopts the concept of state space representation (from now on SSR) in the matching process and proposes five feasible rules to prune the search space. Compared with VF algorithm, the most significant improvement is the strategy of traversing the search tree and the data structure making the algorithm applied to match the graph with thousands of nodes.

The primary idea of the VF2 algorithm is as follows. Given the digraphs $G_1(N_1, B_1)$ and $G_2(N_2, B_2)$, shown in Figure 2, we are looking for the isomorphic mapping between them. Map M is used to express (n, m) , in which n donates a node of G_1 and m donates a node of G_2 . The process of finding the mapping M is described by SSR. Each state s in the matching process is a partial mapping $M(s)$, which is a subset of M . $G_1(s)$ donates the subgraph of the mapping $M(s)$

associated with G_1 , and $G_2(s)$ donates the subgraph of G_2 matched by $M(s)$. $V_1(s)$ and $V_2(s)$, respectively, represent the set of vertices in $G_1(s)$ and $G_2(s)$. $E_1(s)$ and $E_2(s)$, respectively, denote the edge set in $G_1(s)$ and $G_2(s)$. Given the middle state sp , the partial M is as follows:

$$M = \{(n1, m2), (n2, m1), (n3, m3), (n4, m6), (n5, m4), (n6, m5)\}$$

$$M(sp) = \{(n1, m2), (n2, m1), (n3, m3), (n4, m6)\}$$

$$V1(sp) = \{n1, n2, n3, n4\} \quad (1)$$

$$V2(sp) = \{m2, m1, m3, m6\}$$

$$E1(sp) = \{\langle n1, n2 \rangle, \langle n2, n3 \rangle, \langle n3, n4 \rangle\}$$

$$E2(sp) = \{\langle m2, m1 \rangle, \langle m1, m3 \rangle, \langle m3, m6 \rangle\}$$

There are multiple states in the matching process, and state s is converted to another state by adding a pair of new nodes. By adding different pairs of nodes, s is converted to various states. In this way, the new state is described using a tree structure in which parent node represents the original state and the child node represents the new state. In Figure 2, s converts to sq after adding node $(n5, m4)$. Figure 3(a) shows that the node pairs $(n5, m4)$ are just one of many possible ones. Therefore, we need to select the appropriate state by backtracking the search tree. In Figure 3(b), after joining $(n5, m4)$, $G_1(sp)$ and $G_2(sp)$ are successfully converted to $G_1(sq)$ and $G_2(sq)$.

In the matching process, M is obtained by searching the SSR. VF2 algorithm proposes five feasible rules to reduce the time complexity by pruning the search space. According to the proposed rules, the unsatisfied child nodes are removed. The remaining nodes set is called the candidate set $H(s)$, which is traversed in the depth-first order. The pseudocode of VF2 algorithm is shown in Algorithm 1.

The following definitions are given:

- (1) $T_1^{out}(s)$: it denotes a vertex set of G_1 , vertexes of which are descendent vertexes of $G_1(s)$ but not contained in $G_1(s)$.
- (2) $T_2^{out}(s)$: it denotes a vertex set of G_2 , vertexes of which are descendent vertexes of $G_2(s)$ but not contained in $G_2(s)$.

```

Input:  $G_1, G_2$ , State  $s$ , initialized state:  $s_0$ ,  $M(s_0)$  is set empty
Output: The isomorphic map:  $M$ 
(01) PROCEDURE VF2 Match( $s$ )
(02)   IF  $|M(s)| = |G_2|$  THEN
(03)     Successful Match
(04)   ELSE
(05)     Find  $H(s)$  which is the set of possible pairs for  $M(s)$ 
(06)     FOREACH  $h$  in  $H(s)$ 
(07)       IF all rules are satisfied for  $h$  added to  $M(s)$  THEN
(08)          $s' = \text{put } h \text{ into } M(s)$ 
(09)         CALL VF2Match ( $s'$ )
(10)       ENDIF
(11)     ENDFOREACH
(12)   Restore data
(13) ENDIF
(14) END PROCEDURE VF2MATCH

```

ALGORITHM 1: The original VF2 algorithm.

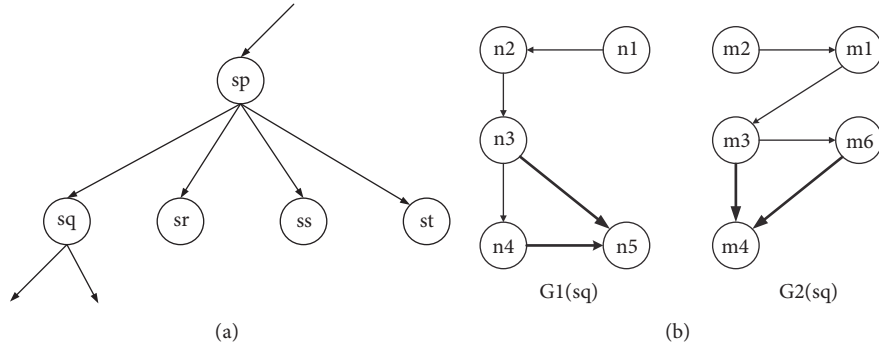


FIGURE 3: SSR state transition diagram.

- (3) $T_1^{in}(s)$: it denotes a vertex set of G_1 , vertexes of which are antecedent vertexes of $G_1(s)$ but not contained in $G_1(s)$.
- (4) $T_2^{in}(s)$: it denotes a vertex set of G_2 , vertexes of which are antecedent vertexes of $G_2(s)$ but not contained in $G_2(s)$.

The steps of selecting $H(s)$ are as follows:

- (1) If $T_1^{out}(s)$ and $T_2^{out}(s)$ are not empty sets, then $P(s) = T_1^{out}(s) * T_2^{out}(s)$.
- (2) If $T_1^{out}(s)$ and $T_2^{out}(s)$ are both empty sets and $T_1^{in}(s)$ and $T_2^{in}(s)$ are not empty sets, then $P(s) = T_1^{in}(s) * T_2^{in}(s)$.
- (3) If $T_1^{out}(s)$, $T_2^{out}(s)$, $T_1^{in}(s)$, and $T_2^{in}(s)$ are empty sets, then $P(s) = (V_1 - V_1(s)) \times (V_2 - V_2(s))$.
- (4) Other conditions prune the state s .

As described above, if one of $T_1^{out}(s)$ and $T_2^{out}(s)$ or one of $T_1^{in}(s)$ and $T_2^{in}(s)$ is an empty set, state s is pruned. For state s , the algorithm needs to check all the candidate nodes (m, n) by the feasibility function $F(s, n, m)$, in which s denotes the

current state, n denotes a vertex of G_1 , and m represents a vertex of G_2 . The return value of $F(s, n, m)$ reflects whether the given node is feasible. If the node is not feasible, the path of it will be pruned.

The feasibility rules are divided into grammatical and semantic. The grammatical rules express the topological structure of the graph, and the semantic ones express the properties of the vertices and edges. In this work, we consider the grammar rules because there are no properties in edges and vertexes of ORG. Therefore, $Fsyn(s, n, m)$ is defined as follows:

$$Fsyn(s, n, m) = R_{pred} \wedge R_{succ} \wedge R_{in} \wedge R_{out} \wedge R_{new} \quad (2)$$

Five feasible grammar rules are defined in $Fsyn(s, n, m)$, in which R_{pred} and R_{succ} are the consistency of $M(s)$. After the candidate node (m, n) is added, R_{in} , R_{out} , and R_{new} are used to prune the search space.

$Pred(G, n)$ denotes the set of the antecedent nodes of n in figure G , and $Succ(G, n)$ denotes the set of the descendent nodes of n in figure G . The algorithm defines $T_1(s) = T_1^{in}(s) \vee T_{out}(s)$, $N_1'(s) = N_1(s) - M_1(s) - T_1(s)$. $T_2(s)$ and $N_2'(s)$ are defined as $T_2(s) = T_2^{in}(s) \vee T_{out}(s)$, $N_2'(s) = N_2(s) - M_2(s) - T_2(s)$.


```

Input:  $G_1, G_2$ , state  $s$ , the initial state:  $s_0$ ,  $M(s_0)$  is empty,  $\lambda$ : Precision control parameters
Output: Isomorphic Mapping
(01) PROCEDURE VF2 Match( $s$ )
(02)     IF  $|M(s)| \geq \lambda|G_2|$  THEN
(03)         Successful Match
(04)     ELSE
(05)         Find  $H(s)$  which is the set of possible pairs for  $M(s)$ 
(06)         FOREACH  $h$  in  $H(s)$ 
(07)             IF all rules are satisfied for  $h$  added to  $M(s)$  THEN
(08)                  $s' = \text{put } h \text{ into } M(s)$ 
(09)                 CALL VF2Match( $s'$ )
(10)             ENDIF
(11)         ENDFOREACH
(12)     Restore data
(13)     ENDIF
(14) END PROCEDURE VF2MATCH

```

ALGORITHM 2: λ -VF2 algorithm (based on VF2 algorithm).Rule 1 ($R_{pred}(s, n, m)$).

$$\begin{aligned}
 & ((\forall n' \in M_1(s)) \cap \text{Pred}(G_1, n) \exists m' \\
 & \in \text{Pred}(G_2, m) \mid (n', m') \in M(s)) \\
 & \wedge ((\forall m' \in M_2(s)) \cap \text{Pred}(G_2, n) \exists n' \\
 & \in \text{Pred}(G_1, n) \mid (n', m') \in M(s))
 \end{aligned} \quad (3)$$

Rule 2 ($R_{succ}(s, n, m)$).

$$\begin{aligned}
 & ((\forall n' \in M_1(s)) \cap \text{Pred}(G_1, n) \exists m' \\
 & \in \text{Succ}(G_2, m) \mid (n', m') \in M(s)) \\
 & \wedge ((\forall m' \in M_2(s)) \cap \text{Pred}(G_2, n) \exists n' \\
 & \in \text{Succ}(G_1, n) \mid (n', m') \in M(s))
 \end{aligned} \quad (4)$$

Rule 3 ($R_{in}(s, n, m)$).

$$\begin{aligned}
 & (\text{Card}(\text{Succ}(G_1, n) \cap T_1^{in}(s)) \\
 & \geq \text{Card}(\text{Succ}(G_2, m) \cap T_2^{in}(s))) \\
 & \wedge (\text{Card}(\text{Pred}(G_1, n) \cap T_1^{in}(s)) \\
 & \geq \text{Card}(\text{Pred}(G_2, m) \cap T_2^{in}(s)))
 \end{aligned} \quad (5)$$

Rule 4 ($R_{out}(s, n, m)$).

$$\begin{aligned}
 & (\text{Card}(\text{Succ}(G_1, n) \cap T_1^{out}(s)) \\
 & \geq \text{Card}(\text{Succ}(G_2, m) \cap T_2^{out}(s))) \\
 & \wedge (\text{Card}(\text{Pred}(G_1, n) \cap T_1^{out}(s)) \\
 & \geq \text{Card}(\text{Pred}(G_2, m) \cap T_2^{out}(s)))
 \end{aligned} \quad (6)$$

Rule 5 ($R_{new}(s, n, m)$).

$$\begin{aligned}
 & (\text{Card}(N'_1(s) \cap \text{Pred}(G_1, n)) \\
 & \geq \text{Card}(N'_2(s) \cap \text{Pred}(G_2, n))) \\
 & \wedge (\text{Card}(N'_1(s) \cap \text{Succ}(G_1, n)) \\
 & \geq \text{Card}(N'_2(s) \cap \text{Succ}(G_2, n)))
 \end{aligned} \quad (7)$$

The above five rules are applied to the subgraph isomorphism pattern. In addition, for isomorphism pattern, “ \geq ” in R_{in} , R_{out} , and R_{new} is replaced by “ $=$ ”. If the newly added node pair is satisfied by the five feasibility rules, the algorithm adds them and continues the searching.

3.5. The Implementation of λ -VF2 Algorithm. In this section, we propose λ -VF2 algorithm based on the environment of Android to detect subgraph isomorphism between the ORG and ORGB. According to Section 3.4, the VF2 algorithm is aimed at isomorphism and subgraph isomorphism. However, for the study of ORG, in the case of subgraph isomorphism, it is still difficult to match the subgraph with the original graph. The reason is that the running time for an app injected with malicious code is not sufficient, which causes the creation of the incomplete references. Therefore, the algorithm needs to be adjusted to relax the matching condition. To relax the matching condition, the algorithm finishes when the matching ratio of vertex reaches a proper threshold.

The threshold $\lambda \in (0, 1)$ is set as the input of the algorithm, which is determined by the user. λ indicates that the algorithm is terminated only when the ratio of matched vertices is bigger than or equal to λ ; the algorithm returns *success*. In this way, the pseudocode of λ -VF2 algorithm is shown in Algorithm 2.

3.6. Performance Analysis. The time and space complexity of VF algorithm is positively correlated with λ . As an input

parameter, λ is independent of the algorithm. In this section, λ is considered as 1 at the worst case.

3.6.1. Time Complexity. Our algorithm is a graph SSR-based isomorphism algorithm. The time complexity consists of two parts: the time of traversing and the processing time for each state.

(i) *Traversing Time.* At best, each state has only one satisfied candidate node; namely, there is no need for backtracking. The total number of states that need to traverse is the number of nodes in given graph. The worst case is that there are no unsatisfied states. In the $d + 1$ th level of the search tree, there are $N(N - 1)(N - 2) \cdots (N - d)$ nodes. And the total number of tree nodes is

$$1 + N + N(N - 1) + N(N - 1)(N - 2) + \cdots + N(N - 1)(N - 2) + \cdots + 2 = 1 + \frac{N!}{(N - 1)!} \quad (8)$$

$$+ \frac{N!}{(N - 2)!} + \frac{N!}{(N - 3)!} + \cdots + \frac{N!}{1!} = 1 + N! \sum_{d=1}^{N-1} \frac{1}{d!}$$

$\sum_{d=1}^{N-1} (1/d!)$ is less than 2. Thus, the total number of sizes is $O(N!)$.

(ii) *Processing Time of Each State.* The processing time for each state consists of three parts: the calculation time T_H of the candidate set $H(s)$, the calculation time T_F of the feasible function $F(s, n, m)$, and the calculation time T_{new} of the new state. The total time of every single state: $T = T_H + T_F + T_{new}$.

T_H : the processing time for each state in the candidate set is constant, and the maximum size of the set is N . Therefore, T_H is $O(B)$.

T_F : in the process of $F(s, n, m)$, each edge costs constant time and the number of edges in the worst case is the number of nodes which is connected to every remaining node. Thus, $T_F = (B)$.

T_{new} : the calculation time of the new status includes the time of $M(s')$, $V_1^{in}(s)$, $V_1^{out}(s)$, $V_2^{in}(s)$, and $V_2^{out}(s)$, in which $M(s')$ is cost constant time. And the other four sets need to iterate over the edges of the newly joined one, which is $O(B)$ at the worst case.

B is the number of edges that a node is connected to. Given a directed graph of N vertexes, the number of edges connected to one given vertex achieves the maximal number of $2 * (N - 1)$. Therefore, $O(B) = O(N)$ in the worst case.

In summary, $T = T_p + T_F + T_{new} = O(N) + O(N) + O(N) = O(N)$.

Final Time Complexity. According to the above analysis, the time complexity of the VF2 algorithm is the multiplication of the two parts.

In the best case, $O(N) * O(N) = O(N^2)$.

In the worst case, $O(N) * O(N!) = O(N * (N!))$.

3.6.2. Space Complexity. The VF2 algorithm adopts the sharing data structure. Thus, the storage space number required

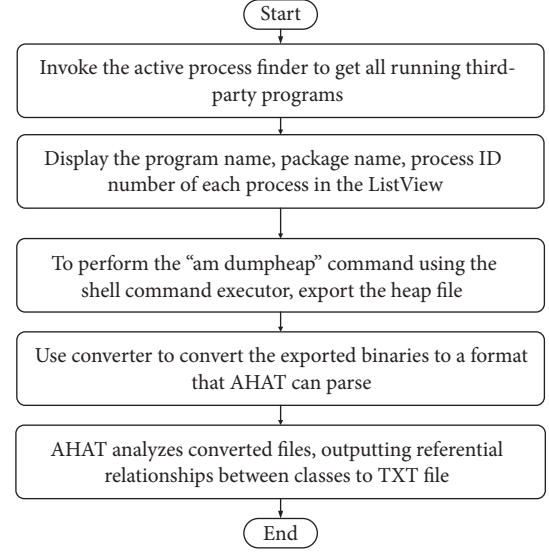


FIGURE 4: Malware detection inspection flowchart.

by each state is constant. The searching process traverses the search tree in the depth-first order, and the maximum depth of the tree is less than N . Therefore, the space complexity is $O(N)$.

4. Framework of Demadroid

Demadroid mainly includes two parts: Android client and PC server. Android client is responsible for extracting data and passing it to the server side, and PC server is responsible for the malware detection.

4.1. Design and Implementation of Android Client. The main function of the Android module is to extract the object reference information from a process. We construct Malware-Detection to analyze the running process (except the system process) and export the dynamic information file for further analysis.

The main components of MalwareDetection include front-end interface, active process finder, shell command executor, Convertor, and AHAT. The extraction flow is shown in Figure 4.

In general, the existing malicious code is embedded in the normal apk. After installation, the malicious code starts with the host app, sharing the process resource in memory. Objects are created in the process, each of which has mutual references with each other. The information we need includes the objects created by the injected process and references between them. We extract the information above in Android client. The reason is that the size of raw memory file is too large. For example, a lightweight app “calculator” generates a memory file of 10 M. There are many processes running in the memory at the same time. Therefore, it is necessary to extract the useful information to reduce the network burden when uploading to PC server.

TABLE 1: The extraction environment of the Android memory data.

| PC OS | Tools used | Virtual Android System Version |
|-----------|-------------------------------|--------------------------------|
| Windows 7 | ADT (Android Developer Tools) | Version 4.0.3 |

| |
|--|
| VersionHead Identifier(4B) File Creation Date(8B) Unit 1 Unit 2 ... Unit n |
|--|

Box 1: Dumpheap file format.

There are three steps in the extraction process. The first step is the acquisition of raw heap information. The second step is to convert the raw memory file format. The third step is to analyze the dynamic information.

4.1.1. The Acquisition of Heap Memory Information Files. The Android SDK provides feature-rich memory monitoring tools, such as dumpheap tools for heap data monitoring. And it is supported by Android 2.3 version or more. To facilitate the analysis, we use AVD to virtualize Android 4.0.3 and successfully extract the heap data of the test process by dumpheap. The data extraction environment is shown in Table 1.

The dumpheap command is in the format of “*am dumpheap PID path*”. We integrate dumpheap into Android program. In the extraction process, we first use the adb tools to obtain the equipment information. After the execution of this command, the heap data of process is saved in files. In this way, a complete file of raw heap information is obtained. This file is binary and cannot be read directly from the contents. Therefore, the format of the binary file needs to be converted.

For example, we start the “calculator” application in the virtual device. With the obtained process ID number, we export the memory raw data of the *Calculator* process by dumpheap.

4.1.2. The Format Conversion of the Memory Information File. The raw memory data is binary and it cannot be analyzed directly. We develop Converter to convert it into an available format.

The analysis tool we propose, AHAT, is based on JHAT, which is used in PC environment. The version of the binary memory file generated by dumpheap is 1.0.3, while the version JHAT can analyze is 1.0.2, and the file format needs to be converted from 1.0.3 to 1.0.2 on Android platform.

The function of Converter is similar to HprofConv tools of SDT, which is used in PC environment. The first step is to analyze the two versions. The binary file format produced by dumpheap is shown in Box 1. The format of the binary file is

| |
|--|
| Type(1B) TimeStamp(4B) Length(4B) DetailInfo(Length * 1B) |
|--|

Box 2: The format of unit.

TABLE 2: New type of 1.0.3 unit.

| Type | Hexadecimal mark |
|-----------------------------------|------------------|
| HPROF_HEAP_DUMP_INFO | 0xfe, |
| HPROF_ROOT_INTERNED_STRING | 0x89, |
| HPROF_ROOT_FINALIZING | 0x8a, |
| HPROF_ROOT_DEBUGGER | 0x8b, |
| HPROF_ROOT_REFERENCE_CLEANUP | 0x8c, |
| HPROF_ROOT_VM_INTERNAL | 0x8d, |
| HPROF_ROOT_JNI_MONITOR | 0x8e, |
| HPROF_UNREACHABLE | 0x90, |
| HPROF_PRIMITIVE_ARRAY_NODATA_DUMP | 0xc3, |

fixed, beginning with a version string, such as “*Java PROFILE 1.0.2*”, followed by the 4-byte ID information, followed by 8-byte file creation date information. After creation date information is the memory data, which is the body of the binary file.

The memory data consists of units. Each of these units stores the information of a Java object. The format of a unit is shown in Box 2. The data structure includes a 1-byte type field, a 4-byte timestamp field, a 4-byte data length field n , and finally the n -byte object information field.

The main difference between the two versions is that the number of types is in Detail Info field. In the old version, there are thirteen types in the Detail Info field. In the new version, nine new types are added, which are shown in Table 2.

The types shown in Table 2 make the information unanalyzable. The solution is to remove the new types, which is irrelevant to our work.

We use the unit types as the member of *Converter* class, which is used in the analysis process to determine whether a given type is useful. Finally, the file is reorganized in the format of the 1.0.2 version.

4.1.3. Extraction of Object and Reference Information. We develop AHAT, a tool used to analyze binary files in Android which is similar to JHAT in PC environment. AHAT mainly consists of four parts: Model, Parser, Util, and external call interface. The relationship between the four modules is shown in Figure 5.

Model. It defines the types (data structures) of all involved objects, and the objects of these data structures constitute a model. There are 29 classes corresponding to object types of Java, the most important of which is the *Snapshot*, the largest unit of the memory snapshot model.

TABLE 3: Classes that need to be filtered out during extraction.

| | | | |
|---------|---------------------|-----------------|----------------------|
| boolean | long | javax.net. | javax.transaction. |
| char | sun. | javax.print. | javax.xml.parsers. |
| float | java. | javax.rmi. | javax.xml.transform. |
| double | javax.accessibility | javax.security. | org.ietf.jgss. |
| byte | javax.crypto. | javax.sound. | org.omg. |
| short | javax.imageio. | javax.sql. | org.w3c.dom. |
| int | javax.naming. | javax.swing. | org.xml.sax. |

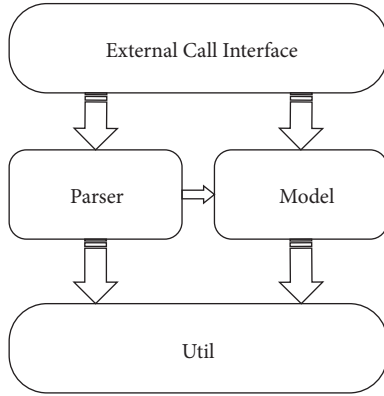


FIGURE 5: The structure of AHAT.

Parser. It is used for reading binary files, analyzing data, and using it with model objects to build a model. Parser consists of 7 classes; the main class is *HprofReader*, used for heap binary parsing.

Util. it is a common toolkit.

External Call Interface. AHAT is responsible for invoking each module to make it work properly. The *activity* class is interacting with the user on Android, so the *main* class is the *MainActivity* class and the *QueryClassInfo* class used to get the referential relationship between the classes.

According to the work process of JHAT, there are four steps in the implementation of AHAT:

- (1) Create: AHAT first creates a snapshot for preparing to store data.
- (2) Read: the *HprofReader* class parses the binary file to obtain the necessary information and builds the *Snapshot* object.
- (3) Resolve: the *Snapshot* object uses the object information to initialize the data structure which includes the reference relationships between classes.
- (4) Query: based on the constructed model, we query the class reference and write it in files.

4.1.4. Important Data Structures and Methods

- (1) *Snapshot* class: It represents a Snapshot of a Java object in the JVM which contains the dynamic object

TABLE 4: The experimental operating environment of AHAT.

| | |
|-----------------|--|
| Device name | Galaxy Nexus 3 |
| Android version | Android4.1.2 |
| Mobile RAM | 1 GB |
| CPU | Texas Instruments OMAP4460, dual-core, Frequency 1228 MHz |

information as well as references between them. The data structures involved are defined in the model module.

- (2) *HprofReader* class: It parses the binary file to extract the memory information of each unit and uses it to build a *Snapshot* object. After this, we initialize the data structure, calculate the specific information of each object, like package name, class name, class ID, class member variable, reference relation between classes, and so on. The above process is the key to dynamic information extraction.
- (3) *QueryClassInfo* class: The function of *QueryClassInfo* class is to extract the references between classes of *Snapshot* object. The variable *referrersStat* in the *process* function is a Hashmap which stores the referenced information of this class and the variable *referrersStat* is used to store the referencing information. All the classes in the memory are obtained by the function *getClasses* of *Snapshot*.
- (4) *PlatformClasses* classes: In the obtaining process of object references, there are thousands of classes returned by function *getClasses*, most of which are *platform-supplied* classes, like the Java Standard API classes, the API classes provided by the Android system, and so on. These classes are irrelevant to our work. What is more, the existence of them can cover the references between the key classes. Therefore, we remove such irrelevant classes (shown in Table 3) by function *PlatformClasses*.

4.1.5. Results of AHAT. The AHAT requires Android 4.0 or more. We test it on Google's Galaxy Nexus 3, of which the environment is shown in Table 4.

The analysis process includes the reading of dumpheap files, binary data parsing, class reference relationship analysis, and the creation of result files. The result is stored in the dumpheap folder of the SD card.

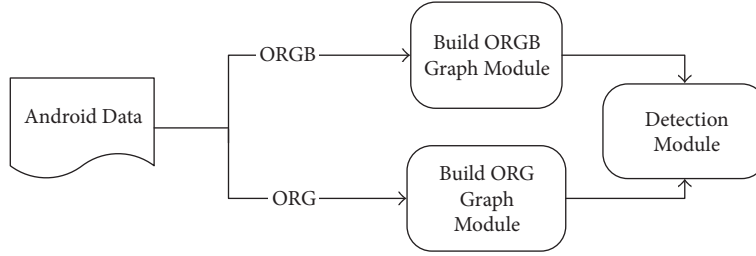


FIGURE 6: The overall architecture of the server.

TABLE 5: Android setup.

| Name | Android System Version | CPU | Phone Memory |
|----------------|------------------------|---|--------------|
| Galaxy Nexus 3 | Android4.1.2 | Texas instrument OMAP4460, dual core, Frequency: 1228 MHz | 1 GB |

4.2. Design and Implementation of Server Side. There are three parts in PC server: the establishment of ORG, the establishment of ORGB, and graph matching. The architecture of PC server is shown in Figure 6. After ORG is created, it is sent to the detection module to match with ORGB by λ -VF2 algorithm.

4.2.1. The Establishment of ORG. ORG is a digraph created by the information obtained in Android client. There is no system class in ORG, in which the nodes represent classes and the edges represent the references between classes. The flow chart of ORG establishment module is shown in Figure 7.

The node ID in the program is a number, and the class name in the file needs to be converted to ID. Thus, we create an index file to assign an ID for each class. In the parsing process, the class name is identified in the index file and added to ORG as a node. When the process identifies the string “*Referrers by type*”, the referencing class is added and the directed edge is established from this node to the referenced node. When the program identifies “*Referees by type*”, it reads the referenced class and adds it to ORG with the directed edge.

4.2.2. The Establishment of ORGB. ORGB is a digraph used to express the feature of malicious code. ORGB only collects the classes of malicious code as nodes, and the class list of malicious code is obtained by manual analysis. The flow chart of ORGB establishment module is shown in Figure 8.

4.2.3. Detection Module. In this part, we propose λ -VF2 algorithm. When the value of λ is 1, λ -VF2 algorithm degrades to the original VF algorithm. In the experiment, the results are different by setting λ with different values. The flow of the detection module is shown in Figure 9.

The program first inputs the value of λ and selects ORG and then matches the selected ORG with every ORGB in the malware library. The matching process will be terminated by a successful match. For the convenience of the experiment, ORG and ORGB are stored in binary file with no attribute of nodes and edges.

TABLE 6: PC setup.

| OS | CPU | Memory |
|-----------|---|--------|
| Windows 7 | Xeon E3-1200 v2, Quad core, 8 threads, 3300 MHZ | 1 GB |

TABLE 7: Number of simulative code samples.

| Origin | Extra Reference | Extra Class | Class Replacement |
|--------|-----------------|-------------|-------------------|
| 4 | 2 | 2 | 2 |

5. Experiments

5.1. Setup. In our experiments, we run the Android apps and extract original data by the tools we developed. We construct ORG and test it with the malware dataset.

(i) *Android Setup.* We extract memory data on a real device. Table 5 shows the experiment environment.

(ii) *PC Setup.* The ORG is sent to PC server. The environment of PC server is shown in Table 6.

5.2. Datasets. We use two kinds of datasets in our experiments, simulative malicious samples, and real malware samples.

5.2.1. Simulative Samples. Each simulative sample is built by manual construction, which consists of two packages. One is malicious and the other is benign. In a given category of simulative malware, the different sample contains different benign packages and the same malicious packages. The advantage of simulating samples is that we can control the scale and operation of malware. In the experiments, we construct 10 simulative samples. The malicious codes in these samples are basically the same. In order to test different effects of VF2-isomorphism, VF2 monomorphism, λ -VF2 isomorphism, and λ -VF2 monomorphism, we adjust the malicious codes to simulative the attacks. Table 7 shows the number of each type in simulative code samples.

- (1) Origin samples: the malicious codes are the same, and the benign parts are different.

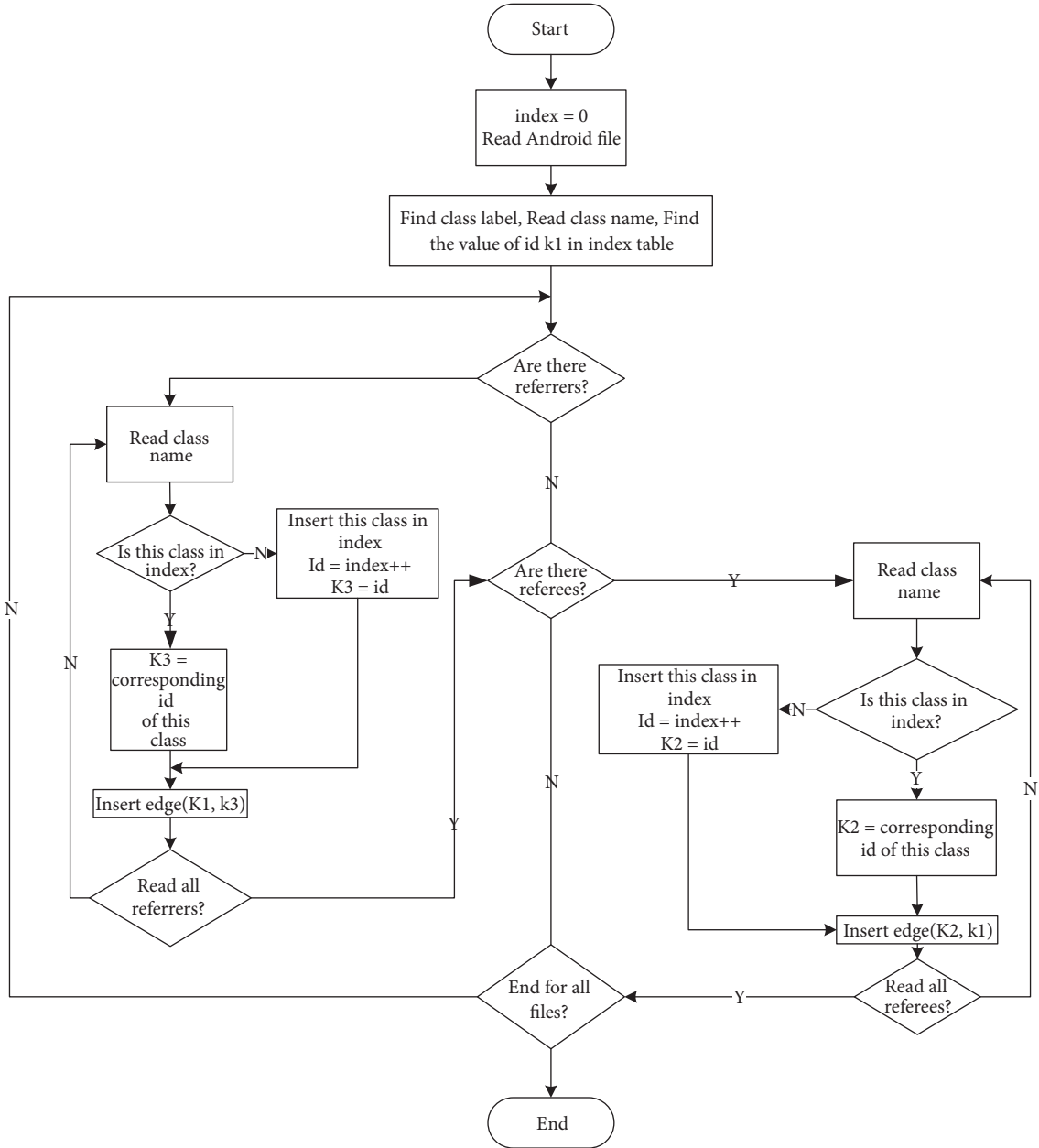


FIGURE 7: The flow chart of the ORG establishment module.

- (2) Extra Reference samples: this kind of samples is simulating the malware which is intended to avoid detection by adding disturbance reference. The classes in malicious codes are identical. However, compared with the original malicious ones, there are several new meaningless references added between classes.
- (3) Extra Class samples: new classes are added based on the original malicious codes to simulate the malicious variations.
- (4) Class Replacement samples: based on the variations of simulative malicious codes, some classes are deleted and some classes are added.

TABLE 8: Number of real code samples.

| ADRD | Bgserv |
|------|--------|
| 22 | 16 |

5.2.2. Real Malicious Samples. We also collect two kinds of real malware which is shown in Table 8.

To extract the ORGB of the given category of malware as the dynamic feature, we select some samples from each category randomly and then analyze them manually.

The APK file is generated from packetized dx tools. We use JD disassembler to reverse the source code to obtain

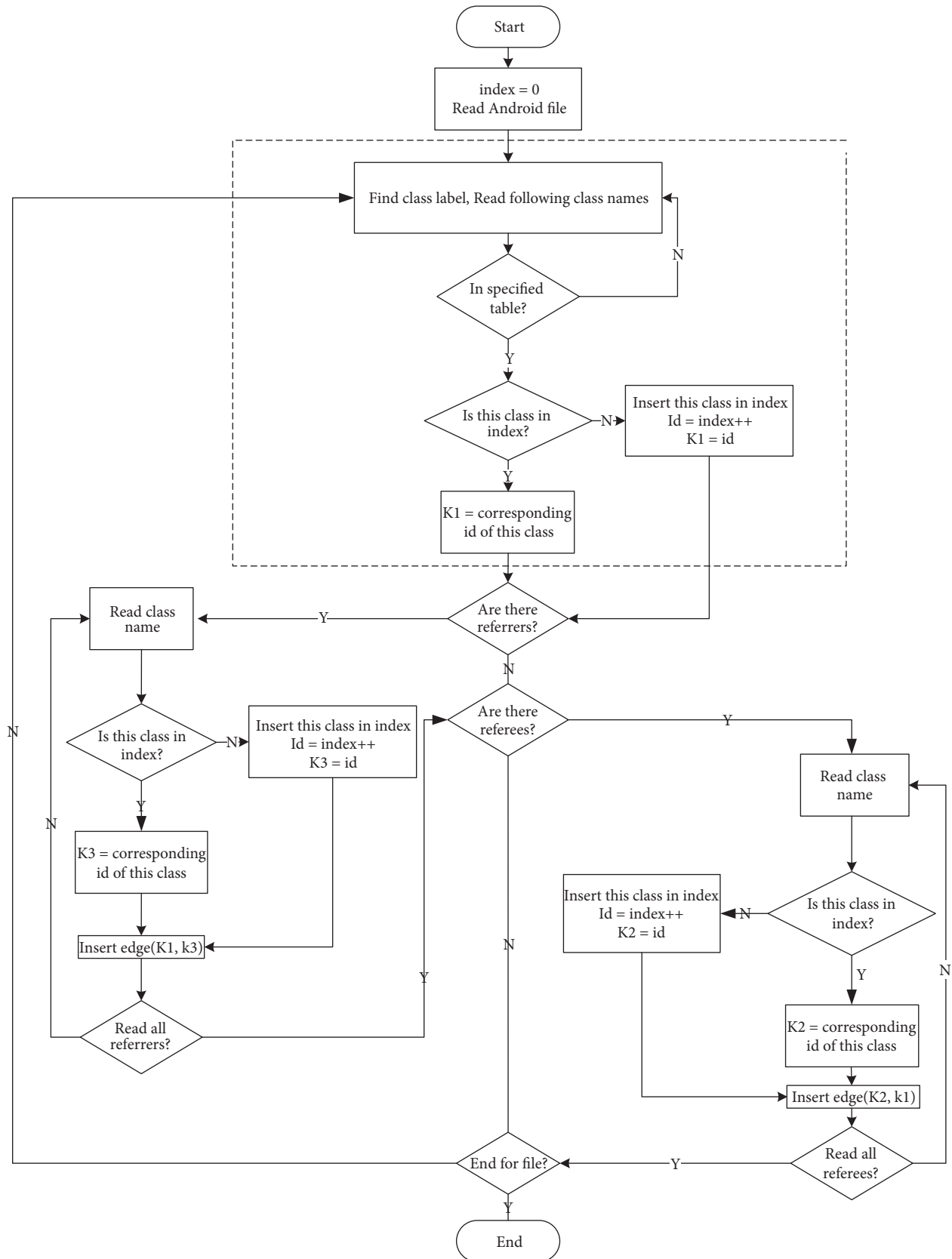


FIGURE 8: The flow chart of the ORGB establishment module.

TABLE 9: Detection results of simulative samples.

| Algorithm | Origin | Extra Reference | Extra Class | Class Replacement |
|-------------------------------------|--------|-----------------|-------------|-------------------|
| VF2 Subgraph Isomorphism | 4/4 | 0/2 | 2/2 | 0/2 |
| VF2 Monomorphism | 4/4 | 2/2 | 2/2 | 0/2 |
| λ -VF2 Subgraph Isomorphism | 4/4 | 1/2 | 2/2 | 1/2 |
| λ -VF2 Monomorphism | 4/4 | 2/2 | 2/2 | 2/2 |

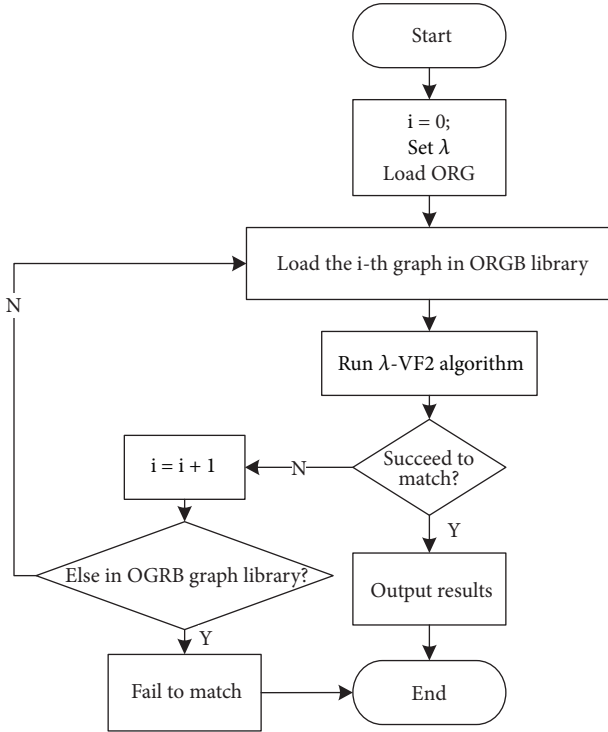


FIGURE 9: Detection module.

the classes. By comparison, we acquire the malicious classes. Classes of malicious codes are generally stored in independent packages, which makes it possible to identify malicious categories manually. Figure 10 shows the file structures in two APKs which contains ADRD malicious codes. Obviously, both apk contains malicious package “xxx.yyy”. In this way, we obtain the list of ADRD.

5.3. Experimental Results on Simulative Samples

5.3.1. Simulation Sample Test Results and Analysis. In our experiments, we first construct ORGB from Origin samples. Then, we construct complete ORG of the 10 samples. Finally, we, respectively, detect ORGB with four kinds of VF2 algorithm. Experimental results are shown in Table 9, where λ is 0.8.

As Table 9 shows, all algorithms can completely detect original malicious codes with new classes added for interference. The reason is that the new classes reflected the new nodes in ORG and ORGB is still a subgraph of ORG. It indicates that our method is effective in the variants added new classes.

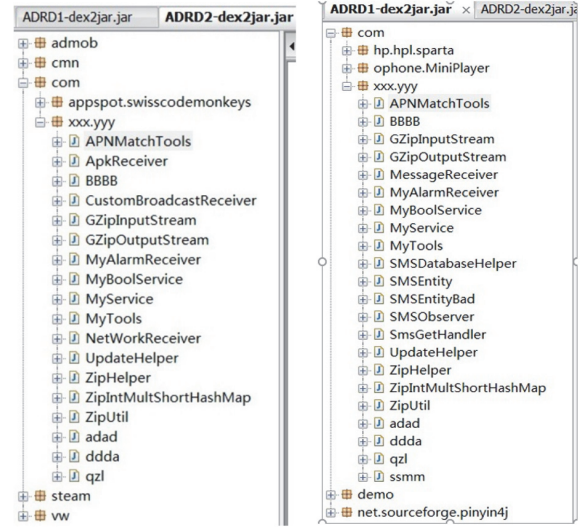


FIGURE 10: Comparison of malicious code classes.

VF2 subgraph isomorphism algorithm is unable to detect the attack of Extra Reference. The reason is that some meaningless references are added, which leads to new edges in ORG. However, subgraph isomorphism requires the complete matching of edges; namely, the new edge is required in both ORG and ORGB.

Extra Reference and Class Replacement are incompletely detected λ -VF2 subgraph isomorphism. This is because the impact of the added references is not completely eliminated and the matching condition is overqualified.

λ -VF2 monomorphism has the weakest constraint and is successful in the four kinds of detection. In practice, even the same kind of malicious codes is not totally identical. And the created objects are different in memory. In consideration of these factors, λ -VF2 monomorphism is the best choice. And the effectiveness needs to be verified on real malware samples.

5.3.2. Confused Variation Detection of Simulative Code Samples. Code confusion is the most common technique used in malware. With code confusion, malware can easily hide the malicious characteristics or generate the variations rapidly, which can avoid static detection.

ProGuard is a famous open source code obfuscation tool, which is integrated into Android. To make it usable, “proguard.config={sdk.dir}/tools/proguard/proguard-android.txt:proguard-project.txt” needs to be added at the end of the properties file.

TABLE 10: Detection results of real samples.

| Algorithm | ADRD | Bgserv |
|-------------------------------------|-------|--------|
| VF2 Subgraph Isomorphism | 1/22 | 1/16 |
| VF2 Monomorphism | 2/22 | 2/16 |
| λ -VF2 Subgraph Isomorphism | 12/22 | 9/16 |
| λ -VF2 Monomorphism | 16/22 | 11/16 |

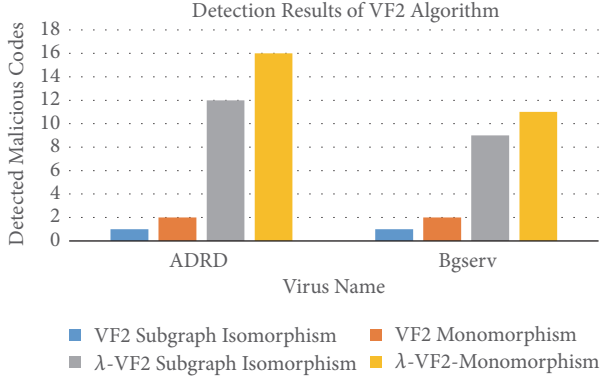


FIGURE 11: Detection results of VF2 algorithm.

In experiments, we utilize ProGuard to obfuscate four Origin simulative samples and regenerate their ORGs. Then, we detect them with the original ORGB by λ -VF2 monomorphism algorithm. Experimental results show that the four ORGs are all matched successfully.

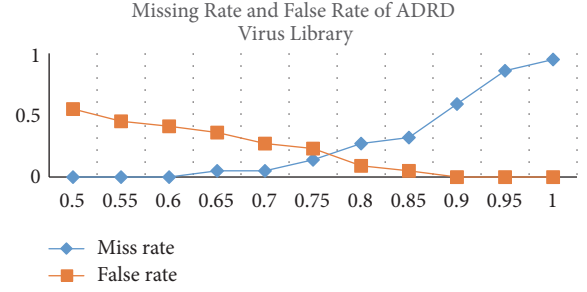
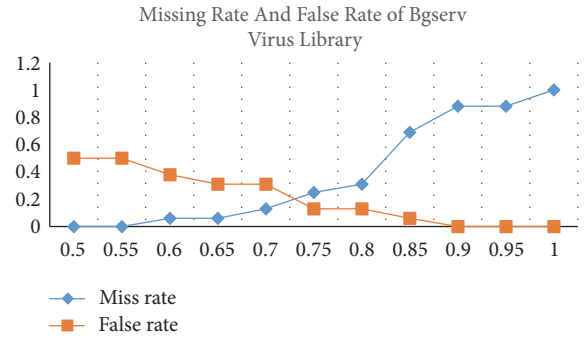
5.4. Experimental Results on Real Samples

5.4.1. Effect of VF2 Algorithm on Malicious Code Detection. The VF2 algorithm is a precise graph matching algorithm, which requires the complete match of the subgraph. This algorithm achieves high accuracy with the low false positive rate. However, the effect of noise leads to the low possibility of complete matching. Thus, the practicability needs to be further tested.

We test the categories of ADRD and Bgserv by VF2 algorithms, and the value of λ is set to 0.8. The experimental results are shown in Table 10 and Figure 11.

As depicted in Table 10 and Figure 11, the success rates of VF2 subgraph isomorphism and VF2 monomorphism are low; the main reasons include the following:

- (1) The feature of malicious codes is not sufficiently extracted because of the difference between samples of each category.
- (2) In the process of extracting, malicious process dynamically creates and destroys classes, which leads to the deficient loading of the key feature in the memory.
- (3) These two algorithms are both precisely matching. And the above two reasons can cause the failure of matching of ORGB and ORG.

FIGURE 12: Results of ADRD Virus Library-Varying λ .FIGURE 13: Results of BGSERV Virus Library-Varying λ .

It can be concluded that the reduction of matching precision can decrease the effect of noise and achieve high matching accuracy.

5.4.2. Effect of λ -VF2 Algorithm Varying Precision. λ -VF2 monomorphism algorithm is effective in real malicious codes. The value of λ affects a lot on matching results. If we decrease the value of λ , the matching precision reduces and the false positive rate increases when it tends to 0. If we increase the values of λ , the matching precision reduces and the false negative rate increases when it tends to 1. Thus, the proper value of λ needs to be tested.

To obtain the false rate when λ decreases, we use a malware group and a benign app group for each test value. And the benign group has the same number of apps with the malware group. λ starts from 0.5 and increases by 0.05 for each group. We obtain the false negative rate from the malware group and the false positive rate from the benign app group. Experimental results are shown in Table 11.

As Table 11 shows, when λ is 0.9, the miss rate achieves 0.5, which impossibly meets the practical needs. When λ is 0.75, the false rate achieves 0.23, which is unsatisfied. Thus, we select the value of λ from 0.75 to 0.85. The variation of miss rate and the false rate is illustrated in Figure 12. Experimental results are shown in Table 12.

As Table 12 shows, when λ is 0.85, the miss rate achieves 0.69, which impossibly meets the practical needs. When λ is 0.7, the false rate achieves 0.31, which is unsatisfied. Thus, we select the value of λ from 0.7 to 0.8. The variation of miss rate and the false rate are illustrated in Figure 13.

TABLE 11: Results of ADRD Virus Library-Varying λ .

| λ | 0.5 | 0.55 | 0.6 | 0.65 | 0.7 | 0.75 | 0.8 | 0.85 | 0.9 | 0.95 | 1 |
|--------------|------|------|------|------|------|------|------|------|------|------|------|
| Total Number | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 22 |
| ADRD | 22 | 22 | 22 | 21 | 21 | 19 | 16 | 15 | 9 | 3 | 1 |
| Normal | 12 | 10 | 9 | 8 | 6 | 5 | 2 | 1 | 0 | 0 | 0 |
| Missing Rate | 0.00 | 0.00 | 0.00 | 0.05 | 0.05 | 0.14 | 0.27 | 0.32 | 0.59 | 0.86 | 0.95 |
| False Rate | 0.55 | 0.45 | 0.41 | 0.36 | 0.27 | 0.23 | 0.09 | 0.05 | 0.00 | 0.00 | 0.00 |

TABLE 12: Results of Bgserv Virus Library-Varying λ .

| λ | 0.5 | 0.55 | 0.6 | 0.65 | 0.7 | 0.75 | 0.8 | 0.85 | 0.9 | 0.95 | 1 |
|--------------|------|------|------|------|------|------|------|------|------|------|------|
| Total Number | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| Bgserv | 16 | 16 | 15 | 15 | 14 | 12 | 11 | 5 | 2 | 2 | 0 |
| Normal | 8 | 8 | 6 | 5 | 5 | 2 | 2 | 1 | 0 | 0 | 0 |
| Missing Rate | 0.00 | 0.00 | 0.06 | 0.06 | 0.13 | 0.25 | 0.31 | 0.69 | 0.88 | 0.88 | 1.00 |
| False Rate | 0.50 | 0.50 | 0.38 | 0.31 | 0.31 | 0.13 | 0.13 | 0.06 | 0.00 | 0.00 | 0.00 |

As observed in the two groups of experiments, as λ rises, the miss rate of malicious codes increases while the false rate decreases. These two parameters are a trade-off. In practice, to guarantee that the miss rate and false rate are satisfied, we set the value of λ according to the needs. From the experiments, it can be concluded that when λ is around 0.85, we can achieve a better performance.

6. Conclusion

In this paper, we present ORG to depict the references between objects allocated in heap memory and extract ORGB as the feature of Android malware from ORG. We propose Demadroid, a dynamic system for Android malware detection. After extracting ORG in memory, Demadroid matches ORG with the ORGB of each malware category by λ -VF2 algorithm. Experimental results demonstrate the effectiveness and efficiency of our algorithm. And Demadroid can effectively resist obfuscated attacks and detect the variants of known malware to meet the demand for actual use.

Our important future work is to take the deeper optimization of the graph match algorithm and the ORG establishment. And we can build a virus library in the cloud and combine the algorithm with cloud computing in the future. In this way, our framework can be improved from efficiency and accuracy in various scenarios.

Disclosure

Professor Hui He and Weizhe Zhang are the corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work is supported by the National Key Research and Development Program of China under Grant no. 2016YFB0800801 and the National Science Foundation of China (NSFC) under Grant nos. 61472108 and 61672186.

References

- [1] W. Zhang, H. He, Q. Zhang, and T.-H. Kim, "PhoneProtector: protecting user privacy on the android-based mobile platform," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 282417, 10 pages, 2014.
- [2] A. Developers, "What is android," 2011.
- [3] A. Azfar, K.-K. R. Choo, and L. Liu, "Android mobile VoIP apps: a survey and examination of their security and privacy," *Electronic Commerce Research*, vol. 16, no. 1, pp. 73–111, 2016.
- [4] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, "The evolution of android malware and android analysis techniques," *ACM Computing Surveys*, vol. 49, no. 4, Article ID 3017427, 2017.
- [5] W. Zhang, X. Li, N. Xiong, and A. V. Vasilakos, "Android platform-based individual privacy information protection system," *Personal and Ubiquitous Computing*, vol. 20, no. 6, pp. 875–884, 2016.
- [6] S. Arshad, M. A. Shah, A. Khan, and M. Ahmed, "Android malware detection protection: a survey," in *Proceedings of the International Journal of Advanced Computer Science and Applications*, vol. 7, pp. 463–475, 2016.
- [7] <http://zt.360.cn/1101061855.php?did=1101061451&did=210467032>.
- [8] P. Palumbo, L. Sayfullina, D. Komashinskiy, E. Eirola, and J. Karhunen, "A pragmatic android malware detection procedure," *Computers & Security*, vol. 70, pp. 689–701, 2017.
- [9] H.-J. Zhu, Z.-H. You, Z.-X. Zhu, W.-L. Shi, X. Chen, and L. Cheng, "DroidDet: effective and robust detection of android malware using static analysis along with rotation forest model," *Neurocomputing*, vol. 272, pp. 638–646, 2018.

- [10] É. Payet and F. Spoto, "Static analysis of Android programs," *Information and Software Technology*, vol. 54, no. 11, pp. 1192–1201, 2012.
- [11] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "Andromaly: a behavioral malware detection framework for android devices," *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 161–190, 2012.
- [12] T. Bläsing, L. Batyuk, A.-D. Schmidt, S. A. Camtepe, and S. Albayrak, "An android application sandbox system for suspicious software detection," in *Proceedings of the 5th International Conference on Malicious and Unwanted Software (Malware '10)*, pp. 55–62, Nancy, France, October 2010.
- [13] K. Luo, "Using static analysis on Android applications to identify private information," Tech. Rep., Dept. of Computing and Information Sciences, Kansas State University, 2011.
- [14] S. Dienst and T. Berger, "Static analysis of app dependencies in android bytecode," Technical Note, 2012.
- [15] L. Batyuk, M. Herpich, S. A. Camtepe, K. Raddatz, A.-D. Schmidt, and S. Albayrak, "Using static analysis for automatic assessment and mitigation of unwanted and malicious activities within Android applications," in *Proceedings of the 6th International Conference on Malicious and Unwanted Software, Malware 2011*, pp. 66–72, IEEE, October 2011.
- [16] F. Di Cerbo, A. Girardello, F. Michahelles, and S. Voronkova, "Detection of malicious applications on android OS," *ICWF*, pp. 138–149, 2010.
- [17] T. Isohara, K. Takemori, and A. Kubota, "Kernel-based behavior analysis for android malware detection," in *Proceedings of the 7th International Conference on Computational Intelligence and Security (CIS '11)*, pp. 1011–1015, IEEE, December 2011.
- [18] A. D. Schmidt, R. Bye, H. G. Schmidt et al., "Monitoring android for collaborative anomaly detection: a first architectural draft," TUB-DAI 8, Technische Universität Berlin-DAI-Labor, 2008.
- [19] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 15–26, October 2011.
- [20] X. Wang, Y. C. Jhi, S. Zhu, and P. Liu, "Behavior based software theft detection," in *Proceedings of the Behavior Based Software Theft Detection*, pp. 280–290, 2009.
- [21] B. D. McKay and A. Piperno, "Practical graph isomorphism, II," *Journal of Symbolic Computation*, vol. 60, pp. 94–112, 2014.
- [22] W.-S. Han, J. Lee, and J.-H. Lee, "TurboISO: towards ultra-fast and robust subgraph isomorphism search in large graph databases," in *Proceedings of the 2013 ACM SIGMOD Conference on Management of Data (SIGMOD '13)*, pp. 337–348, June 2013.
- [23] S. Fortin, "The graph isomorphism problem," 1996.
- [24] J. R. Ullmann, "An algorithm for subgraph isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, 1976.
- [25] D. E. Ghahraman, A. K. C. Wong, and T. Au, "Graph optimal monomorphism algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 4, pp. 181–188, 1980.
- [26] B. D. McKay, "Practical graph isomorphism," 1981.
- [27] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "Fast graph matching for detecting CAD image components," in *Proceedings of the 15th International Conference In Pattern Recognition*, vol. 2, pp. 1034–1037, IEEE, 2000.
- [28] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "An improved algorithm for matching large graphs," in *Proceedings of the 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition*, pp. 149–159, May 2001.
- [29] J. Crussell, C. Gibler, and H. Chen, "Attack of the clones: detecting cloned applications on android markets," *European Symposium on Research in Computer Security*, vol. 12, pp. 37–54, 2012.
- [30] P. Foggia, C. Sansone, and M. Vento, "A performance comparison of five algorithms for graph isomorphism," in *Proceedings of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, pp. 188–199, May 2001.

Research Article

BAVP: Blockchain-Based Access Verification Protocol in LEO Constellation Using IBE Keys

Songjie Wei ¹, Shuai Li ², Peilong Liu ³, and Meilin Liu ⁴

¹School of Computer Science and Engineering, Nanjing University of Science & Technology and State Key Laboratory of Air Traffic Management System and Technology, Nanjing 210094, China

²School of Computer Science and Engineering, Nanjing University of Science & Technology, Nanjing 210094, China

³Shanghai Engineering Center for Microsatellites, Shanghai 201203, China

⁴Shanghai Institute of Satellite Engineering, Shanghai 200240, China

Correspondence should be addressed to Shuai Li; 116106000732@njust.edu.cn

Received 28 December 2017; Accepted 5 April 2018; Published 14 May 2018

Academic Editor: Guojun Wang

Copyright © 2018 Songjie Wei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

LEO constellation has received intensive research attention in the field of satellite communication. The existing centralized authentication protocols traditionally used for MEO/GEO satellite networks cannot accommodate LEO satellites with frequent user connection switching. This paper proposes a fast and efficient access verification protocol named BAVP by combining identity-based encryption and blockchain technology. Two different key management schemes with IBE and blockchain, respectively, are investigated, which further enhance the authentication reliability and efficiency in LEO constellation. Experiments on OPNET simulation platform evaluate and demonstrate the effectiveness, reliability, and fast-switching efficiency of the proposed protocol. For LEO networks, BAVP surpasses the well-known existing solutions with significant advantages in both performance and scalability which are supported by theoretical analysis and simulation results.

1. Introduction

This paper is based on the conference paper [1]. Low-Earth-Orbit (LEO) satellite network systems as represented by the Iridium system and Globalstar system have become one of the most heated areas of research. Because of the low orbits, LEO networks have the advantages of short delay and low path-loss compared with traditional satellite networks. In addition, a constellation of multiple satellites in a LEO satellite network system brings true global coverage and efficient frequency reuse. LEO satellite systems play an important role in mobile satellite communications and are supposed to be one of the most important components in future global communications.

Due to the open nature of satellite networks, communications can be easily intercepted by unauthorized or malicious attackers. Mechanisms for ensuring secure communication within satellite networks are key for achieving security within satellite network systems. In these communications systems, the use of encryption algorithms to maintain confidentiality

is a common and effective method. There is significant difference between satellite networks and terrestrial networks in many respects, such as computing capability, storage space, high packet loss rate, and dynamic topology. Consequently, the terrestrial authentication protocols represented by a series of protocols with certificates are less applicable in such scenarios with satellites. On the other hand, the existing public key infrastructure (PKI) must ensure dependability of a third party such as a certificate authority (CA) in general. Certificates and key management overhead are not negligible. Thus, when considering the design of authentication protocols, we ensure secure communication with concern about computation and storage overhead and the number of steps and nodes involved.

Unlike traditional satellite networks, LEO satellite networks have the characteristics of dynamic topology and frequent connection switching. The authentication protocol running on satellite nodes has to be as light-weighted and cost-effective as possible in premise of ensuring security. This means cryptography used in authentication has to be

carefully selected and customized for satellites onboard. A short response time during authentication is also preferred. However, there are a lot of concerns within the centralized authentication protocols in satellite network, such as complex computation, central bottleneck, and long response time, which make the above desires not easily achievable. This paper proposes a Blockchain-based Access Verification Protocol (BAVP) by combining identity-based encryption (IBE) and decentralized blockchain technology. IBE brings in the advantage of fast key generation with specified identity string provided by users, which eliminates the cost of certificates used in traditional authentication protocols. Blockchain contributes to the decentralizing of both data storage and computation.

2. Related Work

Regarding the related literature on centralized authentication protocols used in existing satellite network, Cruickshank proposes an authentication protocol [2] that uses asymmetrical encryption algorithms. However, the operations involved in his protocol are too complicated to implement. Hwang et al. redesign the authentication protocol without a public key cryptosystem [3], but the shared secret key still needs to be updated every time when a user is authenticated. Y. F. Chang and C. C. Chang propose a mutual authentication protocol that requires only XOR and hash function [4], where, during every authentication procedure, a network control center (NCC) need not generate a private key and a temporary identity for user. However, the NCC is involved in every authentication session as critical bottleneck and single-point-of-failure resource which may bring in higher delay during authentication. The performance of the authentication protocol is restricted by NCC. Zheng et al. propose an authentication protocol avoiding these weaknesses by involving a gateway in authentication [5]. Their proposed protocol involves not only users and satellites but also the gateway and NCC during authentication. The number of interactive steps is inflated resulting in a variant response time of authentication. Lin's paper compares and summarizes the characteristics of symmetric encryption, asymmetric encryption, and the certificate system used in satellite network [6].

Additionally, traditional centralized authentication protocols are designed mainly for MEO (Medium Earth Orbit) and GEO (Geosynchronous Earth Orbit). There is less consideration on distributed handover authentication which is unavoidable in LEO satellite networks with frequent link switching and narrow single-satellite coverage. By simply applying the existing centralized authentication protocols in LEO satellite networks, each handover authentication in a LEO satellite network requires a new complete authentication. This magnifies the disadvantages with these protocols discussed above and thus is inappropriate for LEO satellite networks.

There are several schemes focusing on LEO satellite network as noted in papers [4, 7, 8]. In paper [7], the author proposes an efficient and secure anonymous authentication scheme that requires only XOR and hash function and improves the disadvantages such as user's privacy not being

kept confidential compared to paper [4]. However, it still has the NCC bottleneck during authentication. Wu et al. propose a lightweight authentication and key agreement (AKA) scheme [8] based on the synchronization mechanism of user's temporary identity which fixes the security problems found in paper [9]. All these papers utilize the XOR and hash function for efficient computation, but none of them is optimized for LEO satellite network with NCC still involved.

In summary, PKI is still the most fundamental for implementing key management and not appropriate for LEO with resource constraint. In addition, referring to decentralized authentication protocol used in satellite network, previous researches are relatively lacking. In other resource constraint scenarios similar to satellite networks, such as wireless sensor networks, the authentication protocols investigated intensively focus on mainly cluster and mostly centralized ones.

3. Protocol Design

In the proposed Blockchain-based Access Verification Protocol (BAVP) for LEO authentication, Key Generation Center (KGC) generates public and private keys of all roles (users and satellites) with its private key and these roles' identities. Meanwhile, based on blockchain, a trust chain consisting of KGC, satellites, and users is the core base for rapid handover authentication. With distributed storage in blockchain, this protocol records users' registration, cancellation, login, logout, handover, and other related logs as plugin.

Authentication is divided into two parts: access authentication and handover authentication. During access verification, users and satellites can implement mutual authentication through their public and private keys, and a user's authority is checked against his token. Meanwhile, the relevant authentication logs are recorded in a form of blocks which would be merged and distributed between satellites and the KGC. We describe the logical structure of this system as in Figure 1. A satellite in each orbit is selected as a logical root responsible for the interaction of blocks with KGC. This logical structure is also the basis of blocks' merging and distribution. Before presenting the detailed design, we first briefly review IBE and blockchain technology as background knowledge.

3.1. Background

3.1.1. Identity-Based Encryption. In IBE [10], a user's public key can be derived directly using his unique identity string, such as a phone number and email. IBE eliminates the computation and storage overhead with certificates. In this way, we can create the mapping between identity and public key.

IBE requires a trusted third-party KGC to provide key generation services for different roles in this system. When registering, a user needs to provide his identity to the KGC; then the KGC uses its private and public key together with related system parameters to calculate a pair of public and private keys for this user and also securely transmit them to the user. When sending confidential information, a user needs no certificate but the public key which corresponds

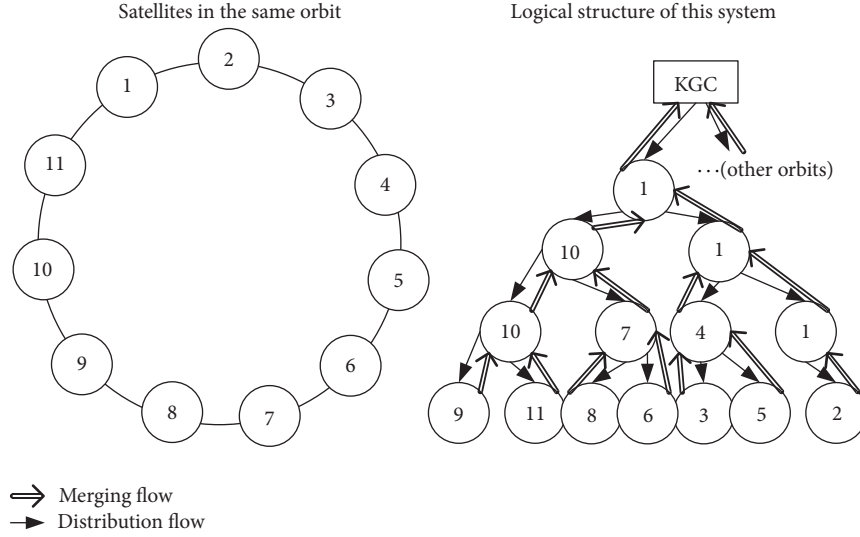


FIGURE 1: Logical structure of this system.

to the receiver's identity in order to encrypt messages before sending.

The most efficient IBE schemes are based on bilinear pairings of elliptic curves, and currently IBE based on pairing is mainly divided into three categories: exponent inversion, full domain hash, and commutative blinding. The full-domain-hash mechanism requires much computation for the mapping between user's identity and a point on elliptic curve, which is not suitable for resource constrained scenarios such as satellite networks. Thus, in a scenario with limited computing power like in satellite networks, the other two schemes are more suitable to be adopted.

3.1.2. Blockchain. Blockchain [11] is the underlying technology that supports Bitcoin. It is essentially a distributed ledger secured by cryptography. Its core strength is that trust is built among distributed nodes and data ensured for integrity without being tampered or forged. Furthermore, blockchain supports customization with smart contracts according to diverse demands.

Data integrity and distributed consensus on trust are the two main advantages of blockchain. The former is guaranteed when each node in the network stores a complete copy of data. And the latter primarily depends on the effectiveness of consensus mechanism with no need for Trusted Third Party (TTP) among nodes. According to different scenarios, blockchain can be classified into three types, namely, public blockchain, private blockchain, and consortium blockchain. The major differences are found in their adopted consensus mechanisms. In the case of LEO satellite network system, consortium blockchain would be more suitable in terms of architecture and various demands like being controllable and manageable. Fabric (a consortium blockchain platform) supported by Hyperledger (a global open source collaboration hosted by the Linux Foundation) is a representation of consortium blockchain with a modular architecture delivering high degrees of confidentiality,

resiliency, flexibility, and scalability. Additionally, there are also some new blockchain technologies emerging like IOTA which takes directed acyclic graph (DAG) instead of linked list as its underlying architecture. Generally speaking, the most popular public blockchain platforms are still Bitcoin and Ethereum. Blockchain technology is still under continuous development and evolution.

3.1.3. Smart Contract. Smart contract is a program protocol intended to verify, facilitate, or enforce the performance of a contract. In this paper, smart contract refers particularly to a contract program running on blockchain as the greatest achievement in blockchain 2.0. Taking Ethereum as an example, smart contract is implemented by EVM (Ethereum Virtual Machine) which is Turing-complete. When a smart contract being programmed by solidity or other smart contract programming languages and deployed on blockchain, it is encoded as EVM bytecode and executed by all mining full nodes. Full node refers to those with a complete copy of data of the blockchain while light node refers to nodes with only partial data in the blockchain.

Due to its programmability, atomicity, consistency, and unambiguity, smart contract contributes greatly to blockchain technology. Users can verify the correctness of smart contract by comparing the bytecode of source code provided by promulgator with the bytecode stored in blockchain. And access control is supported based on accounts within smart contract. Accordingly, smart contract can implement specific business logic on blockchain which makes blockchain more promising and practical in various applications.

3.2. BAVP Principles and Processes. BAVP has two major parts: key management implemented with IBE; authentication and records of related logging which is based on both blockchain and IBE. In describing this protocol, we use the symbolic conventions as shown in Table 1.

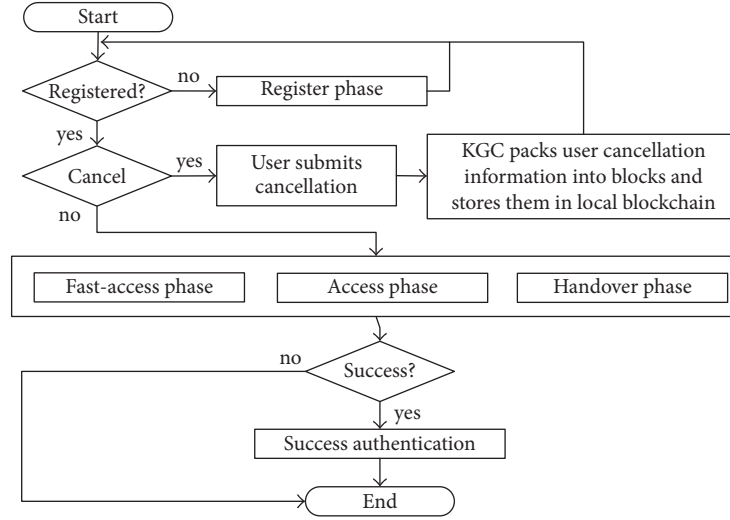


FIGURE 2: Authentication control flow of the proposed protocol.

TABLE 1: Symbols and meanings.

| Symbol | Meaning |
|----------------|-------------------------------|
| ID_A | User A's ID |
| ID_S | Satellite S's ID |
| P_A | User A's public key |
| P_S | Satellite's public key |
| P_{KGC} | KGC's public key |
| d_A | User A's private key |
| d_S | Satellite S's private key |
| d_{KGC} | KGC's private key |
| $Encry_x()$ | Encryption with x as key |
| $Time$ | Time of handover |
| $U_authority$ | User's authority |
| $Start_time$ | Authority's beginning time |
| $Stop_time$ | Authority's ending time |
| XX_Sign | XX's signature |
| $Auth_{Token}$ | User's authorization token |
| $UserInfo$ | User's related info |
| $Service$ | Service that user applies for |
| $result$ | Result of authentication |
| $Sign_x()$ | Signing with x as key |
| $Splace$ | Place of handover |

When explaining the principles of each phase, all messages included in this protocol are timestamped by default, and nodes receiving the messages always check the timestamp. The BAVP control procedure is shown in Figure 2.

3.2.1. Registration Phase. A KGC is a trusted authority which is responsible for calculating a user's public key, private key, and user token for authority. A registered user is allowed to access the satellite system at any time during the token's period of validity. An authorized user submits his identity to KGC to obtain a pair of public and private keys

calculated by the KGC, together with a token signed by the KGC.

The calculation is as follows: user A provides his identity ID_A (such as *user: Alice@gmail.com*, where *user* means the role of user). KGC uses hash function and P_{KGC} to calculate P_A . Next, the KGC calculates d_A with d_{KGC} . Satellites register themselves in the same way before issuance.

Meanwhile, KGC constructs user token of A and signs it with d_{KGC} . And $ID_A \parallel U_authority \parallel Start_time \parallel Stop_time \parallel KGC_Sign$ is the format of $Auth_{Token}$ where KGC_Sign means signature of the first four fields in this token. After finishing, KGC returns the pair of public and private keys, along with the token, to user A safely (e.g., via secure email). Afterwards, KGC packs this user's registration log into blocks which would be stored in local blockchain. At this point, user A has completed the steps necessary for accessing the satellite system. The diagram of the registration phase is shown in Figure 3.

3.2.2. Access Authentication Phase. The access authentication phase is shown in Figure 4, and the four steps are as follows:

(a) When user A wishes to access satellite S , he first checks the identity of S and then uses the hash function to calculate P_S with P_{KGC} . Afterwards, A sends his identity to S .

(b) While receiving this message, S checks the identity of A and searches for latest cancellations to check the validity of A . Then S calculates P_A accordingly, generates random number r together with session key k , and sends $m1$ to A as follows:

$$m1 = Encry_{P_A}(r, k, timestamp, Sign_{d_S}(r, k, timestamp)). \quad (1)$$

(c) After receiving this message, A decrypts it with d_A , verifies the signature of r and k , and then saves them. Thereafter, A sends $m2$ to S as follows:

$$m2 = Encry_k(r, Auth_{Token}, Service, UserInfo). \quad (2)$$

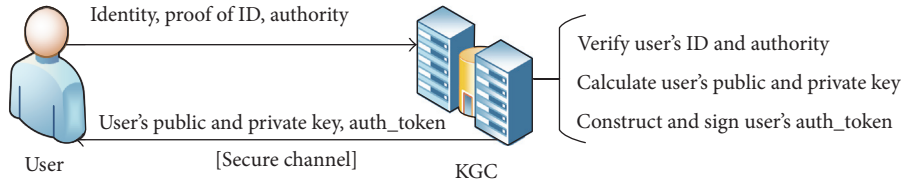


FIGURE 3: Diagram of registration phase.

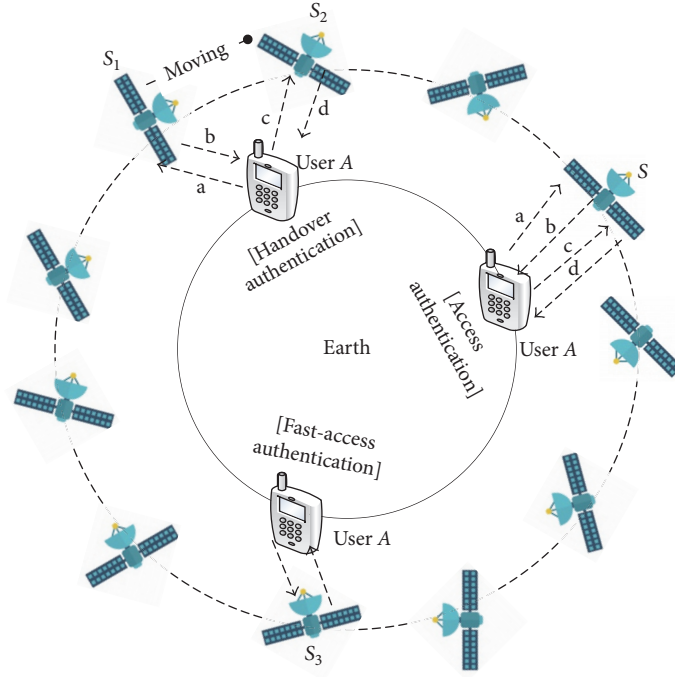


FIGURE 4: Diagram of the protocol.

UserInfo contains the location, time, and *A*'s identity when authentication starts.

(d) While receiving this message, *S* decrypts it with *k*, verifies the correctness of *r*, and searches for the latest cancellations to verify the validity of current user. If *A* is valid, then *S* verifies the signature of *A*'s *Auth_{Token}* with *P_{KGC}*. The session key *k* uses symmetric encryption, such as the Rijndael algorithm. Next, *S* checks whether *ID_A* in *Auth_{Token}* is consistent with the identity provided at the beginning or not.

With all the steps above without mistakes, *S* allocates the resources necessary to establish a secure connection with *A* and provides service according to the authority and expiration time in *Auth_{Token}*. Moreover, *S* packs *A*'s access log which contains *UserInfo* mainly and then stores it into *S*'s local blockchain. Otherwise, *S* disconnects from *A*.

The essence of this phase is to accomplish mutual authentication by IBE. A user needs not store the public key of each satellite in advance. Instead, only through the broadcasted network identification of a satellite, each user can calculate the corresponding public key directly. Authentication security is ensured of IBE. During authentication, a session key

is negotiated, and a secure channel is established after each successful authentication.

3.2.3. Fast-Access Authentication Phase. Once a new user is successfully authenticated, his information would be stored in the access satellite. With data traceability in blockchain, when this user reconnects a satellite for service again, he only needs to send *m3* to the satellite, calculated as

$$m3 = ID_A, Service, ID_{S_3}, timestamp, \quad (3)$$

$$Sign_{d_A} (ID_A, Service, ID_{S_3}, timestamp).$$

S₃ stands for the satellite user *A* wants to access. Next, after receiving this message, *S₃* calculates *P_A* according to *ID_A*, verifies the signature, and then checks if *ID_{S₃}* in this message corresponds to its own. If there is no mistake, then the satellite can search for data related to *A* in its local blockchain, return a new session key which is signed with *d_{S₃}* and encrypted with *P_A*, and provide relevant service according to the relevant data.

Using this procedure, users can access satellites efficiently. The search time is $\log_2(n)$. However, if the satellite being

TABLE 2: Comparisons in authentication phase.

| | Yoon et al. | BAVP: access | BAVP: fast-access | BAVP: handover |
|-------------------------|-------------|--------------|-------------------|----------------|
| Hash operations | 2/4 | (1)/1 | (1)/1 | -/-/2 |
| MAC operations | 2/2 | - | - | - |
| Symmetric operations | - | 1/1 | -/- | -/-/- |
| Asymmetric operations | - | 1/1 | 1/1 | -/-/-1 |
| Signing operations | - | -/1 | 1/1 | 1/1/1 |
| Signature verifications | - | 1/1 | 1/1 | 1/-/2 |
| Communication levels | 2 | 1 | 1 | 1 |
| Authentication center | NCC | None | None | None |

() means only needed for the first time while users can cache satellite public key afterwards; x/y : x means the side of user; y means the side of satellite; $x/y/z$: x means the side of user; y means the side of first satellite; z means the side of second satellite.

accessed is not in the same orbit as the original satellite where the user is previously authenticated, then the user cannot take this fast-access way due to the lack of related data in this current satellite. User who needs the fast-access convenience should access at least one satellite in each orbit previously through regular access authentication procedure.

3.2.4. Handover Authentication Phase. The handover authentication phase is illustrated in Figure 4, and the four steps are explained as follows:

(a) Through the secure channel, user A informs the satellite (called S_1) of his leaving information including ID_A and ID_{S_2} .

(b) While S_1 receives such messages from A , it checks whether the satellite that A wants to switch to is a neighbor or not. For neighbor, S_1 will pack A 's handover log as ($Stime, Splace, Service, ID_{S_1}, ID_{S_2}, ID_A$) into block and store this in its local blockchain. The handover log can also be extended according to user needs. Instantly, S_1 calculates and returns $m4$ to A as

$$m4 = ID_{S_1}, ID_{S_2}, ID_A, timestamp, Service, \quad (4)$$

$$Sign_{d_{S_1}}(ID_{S_1}, ID_{S_2}, ID_A, timestamp, Service).$$

(c) After receiving $m4$, A disconnects from S_1 , signs this message, and sends it to S_2 .

(d) Subsequently, S_2 checks the timestamp of the message received from A and also checks out whether S_1 is its neighbor. If not, S_2 denies A 's request. Otherwise, S_2 calculates P_{S_1} and P_A to verify the signature in this message. When verification succeeds, S_2 searches for the latest cancellations to check the validity of A . If A is valid, S_2 returns a new session key signed with d_{S_2} and encrypted with P_A to A . Later, S_2 officially allocates relevant resources and establishes secure connection with A by this new session key. Meanwhile, S_2 packs A 's handover log which depends on packing the received message mainly into blocks and stores this in its local blockchain.

Next, A decrypts the message received from S_2 with d_A . Then, A verifies the new session key's signature and continues to obtain service through new secure channel between him and S_2 . If any step goes wrong, S_2 disconnects from A .

The core principle of implementing fast handover is its utilization of a trust chain consisting of satellites, users, and KGC. This also brings in consensus among all satellites. When a user is successfully authenticated by passing the check on one satellite in this system, other satellites should recognize the result of authentication as trust.

When it is time to synchronize data (depending on the update interval), each satellite sends its own latest blocks (i.e., blocks that have not been sent out) to adjacent nodes according to the logical organization of the constellation. KGC or satellites would merge these blocks received from other nodes with their own blockchain on the basis of timestamp. If the amount of data at satellite side reaches the threshold, each satellite removes those blocks in accordance with predefined rules, to keep only the latest and mostly queried records.

When a user cancels his identity, KGC packs the user's cancellation record into a block and stores the block in its local blockchain database. Blocks containing the newest cancellation records are periodically or proactively synchronized with P2P distribution as in a typical blockchain.

Regardless of merging or distribution, once a node receives blocks, it verifies the signature of blocks and then integrates these blocks with its local blockchain. The block structure in this protocol is consistent with blockchain. As for re-registration, a user should cancel his original identity and register with a new identity in the same way described in registration phase.

3.3. Performance and Advantages Analysis. As a theoretical analysis of the computational costs required in this proposed protocol, taking symmetric encryption/decryption as P , asymmetric encryption/decryption with IBE as E , signing as N , and signature verification as V , the access authentication phase requires $R_a(2P, 2E, 1N, \text{ and } 2V)$. The fast-access authentication phase costs only $R_f(2E, 2N, \text{ and } 2V)$, and the handover authentication phase needs $R_h(1E, 3N, \text{ and } 3V)$. A comparison of authentication methods between Yoon et al.'s scheme and the protocol proposed in this paper is shown in Table 2.

Since Yoon et al.'s scheme [7] is far superior to those proposed in related works as shown in their paper, Table 2 shows the comparison between Yoon's protocol and the proposed BAVP. As there are only hash and mac operations involved

in Yoon et al.'s proposed scheme, this protocol appears less efficient in computation costs by comparison. Nevertheless, it is not only computation costs that decide whether an authentication protocol is efficient or not. Other factors like communication levels and existence of an authentication center would also affect the efficiency of authentication protocol. As mentioned in Section 2, in Yoon et al.'s scheme, NCC is still involved in authentication which may be the bottleneck of this whole authentication system. Meanwhile, there are two communication levels (user \leftrightarrow satellite and satellite \leftrightarrow NCC) during authentication in Yoon et al.'s scheme, while there is only one communication level with users and satellites in this proposed protocol. And considering the LEO satellite network that has the least network delay (10 ms–40 ms), the forward and backward delay of the extra communication level would bring at least 20 ms for response time of authentication protocol, which is far greater than the time for one operation of asymmetric encryption/decryption (in simulation environment with IBE, it is about 1.5 ms).

From the analysis, we can conclude that BAVP has the following extra merits:

- (1) With IBE, this protocol eliminates certificate cost.
- (2) Using IBE and blockchain, decentralized access authentication and fast handover among satellites can be implemented.
- (3) Based on the trust chain consensus, the system stores information about users and satellites using blockchain technology which ensures the accuracy, completeness, consistency, and traceability of data within the block.
- (4) Auditing is also made possible for protection of network resources and the implementation of security policies by unforgeable logging in blockchain.

3.4. Security Analysis. In the case of common attacks such as data tampering, eavesdropping, replay attacks, and man-in-the-middle attacks, this protocol has intrinsic resistance.

Key Security. A malicious attacker cannot get the plaintexts from the ciphertexts obtained by eavesdropping or sniffing, as long as he cannot get the private key of any user or satellite. Attack cannot tamper with the message, which is based on the security of IBE and AES algorithms. Session security after successful authentication is ensured by session key. Session key negotiation is secured by private keys of users and satellites. As clarified in the previous four sections, private keys of users and satellites are not included directly in various authentication messages, which means these keys cannot be obtained by eavesdropping. When the KGC is credible (keeping d_{KGC} secure and not storing or calculating user private keys illegally after users registered), users and satellites private keys are only known to themselves, which means users themselves are essentially responsible for security of their private keys.

Replay Attacks. The protocol uses timestamps, which can resist such attacks effectively. If there is an attacker who copies an encrypted message by eavesdropping and sends it at another moment, the receiver satellite will reject it after validating the timestamp in the message. Moreover,

the random number r during access authentication phase actually implements a challenge/response method, and also during other phases, the attacker will fail to get session key without possessing private key of the user relevant with the message he replayed; therefore, the satellite will not allocate related resources officially. Thus, this BAVP protocol is resistant to replay attacks.

Man-in-the-Middle Attacks. The man in the middle cannot register with the role of a satellite or the role of an existing user in the system. Therefore, he cannot impersonate any existing role in this system. With IBE, user's identity and relevant public key are bound together, and the receiver can find out whether a message is signed by a specific user. An attacker cannot get the KGC's private key or those of satellites or registered users. Therefore, he cannot disguise himself as any role in the system in order to conduct man-in-the-middle attacks.

Impersonation Attacks. An attacker may attempt to impersonate an authorized user by forging an authentication request. As the first response to such authentication request from satellite during all three kinds of authentication phases should be encrypted using this user's private key, the attacker must know the exact content of right private key in order to be authenticated. However, he has no feasible way to know this private key. The attacker cannot even construct such authentication requests as he has no ability to forge the valid signature of an authorized user during fast-access authentication or handover authentication. At satellite side, if there is an attacker, who attempts to impersonate a satellite, he would fail to forge a valid signature for the session key without possessing the right satellite private key. Even if he replays a previous valid response, he would fail in the next steps of the authentication process due to the check of valid timestamp and the inability to decrypt the session key. Thus, the proposed protocol is secure against impersonation attacks.

Denial-of-Service Attacks. This protocol firstly checks whether the identity of current user is valid and then returns a response which is encrypted with the public key relevant to the identity during authentication. If there is a denial-of-service attack, it would not continue because the attacker has no corresponding private key, which means the satellite would not allocate relevant resources for service and the secure channel between this attacker and the satellite would not be established. Thus, this protocol can resist denial-of-service attacks.

Also, there is no threat of an attack using stolen verification tables or smart cards, as BAVP does not use verification tables or smart cards. Meanwhile, blockchain can ensure accuracy, completeness, consistency, and traceability of data which makes authentication more efficient and more secure. However, the KGC must be completely trusted as required by IBE, which may have potential safety problems hidden within it. It is reasonable to assume that KGC is trustworthy since users must register at KGC with their information to obtain services.

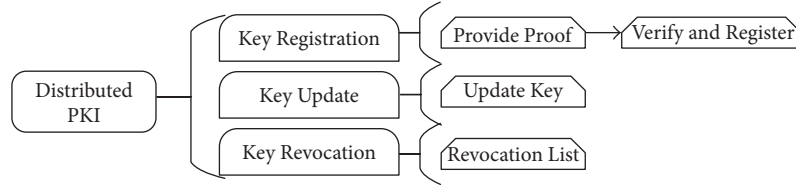


FIGURE 5: Structure of DPKI.

4. Distributed PKI

In the proposed protocol, although KGC is not involved during authentication, it is still the center for key management and is able to calculate all private keys of users. Once it is hacked, the security of the whole system would be threatened. In spite of some solutions that have partially solved this, there are no real all-around solutions. For example, in paper [12], a method based on (n, t) threshold secret sharing cryptography is designed to avoid this problem. The user's private key is split into n pieces and these key fragmentations are stored in different key privacy authority (KPA). Users only need to apply key fragmentation towards enough KPAs, and then they can restore their private key. Thus, this method can avoid the threat brought by centralized key management. Nevertheless, this solution brings additional costs for construction of KPAs, and also the number of KPAs should be large enough under individual owners for security. There are still concerns about KPA mechanism. For example, these KPAs need to take different strong safeguard procedures in order to increase the difficulty of breaking this system. Actually, IBE establishes mapping relations between identity and public key through mathematical methods which avoid the use of certificates, while we can realize this kind of mapping relations through smart contract on blockchain 2.0 like Ethereum. With such thought, we are actually building a distributed PKI (called DPKI).

4.1. Structure of DPKI. There are mainly three functional parts of DPKI: key registration, key update, and key revocation. The structure of DPKI is shown in Figure 5.

4.2. Methodology. DPKI is specifically built using smart contract with blockchain 2.0. Blockchain as a robust P2P network is able to ensure correctness of data stored in it. Thus, making centralized PKI distributed, which can overcome many weaknesses of traditional PKI, becomes possible. This section has the following structure: the first part explains how key registration works, followed by the principles of key update and key revocation and also the code template of DPKI smart contract.

4.2.1. Key Registration. For traditional PKI using certificates, users need to provide proof of their identity to get a valid certificate authorized by a trustworthy CA. With IBE, users need no certificates, but they simply provide a related identity string which can be an email address, ID number, or other strings, and then KGC calculates their private keys which are sent to users safely thereafter. In DPKI, users also submit

proof for their identity and the authority they need. Users generate their public and private key pairs with any kind of asymmetrical encryption algorithm by themselves and then register their public key together with the standard name of algorithm used by invoking smart contract. After this, the administrator of the LEO system checks whether the identity is valid and afterwards passes their registration by invoking pass function of smart contract provided that nothing is wrong. In the satellite scenario, the asymmetrical encryption algorithm used should be limited to several specified algorithms with consideration of resource constraints. The principles are shown in Figure 6.

4.2.2. Key Update. Out of security considerations, users should update their key pairs periodically. With a securely kept password, a user can update his key pairs proactively. During key registration, account address, public key, identity string, and algorithm used for key generation are bound together. Therefore, users willing to update their key pairs are able to do so through the smart contract update function at any time. The key update principles are shown in Figure 7.

4.2.3. Key Revocation. Generally, users need not revoke their key pairs. If their private keys are lost or stolen, they can generate new public and private key pairs and then update their key using the key update method. Nevertheless, when the passwords of users' blockchain accounts are lost or stolen, the users lose all control of key update and revocation. Assuming that there is one user A whose blockchain account is lost or stolen, he can revoke his original identity by submitting relevant proof and then the administrator checks validity of this proof. If this proof is right, the administrator adds A 's original blockchain account to revocation list. Also, the administrator re-register A 's new account with original authority and relevant remaining time after A executes key registration with a new blockchain account. The principles are shown in Figure 8.

4.2.4. Smart Contract of DPKI. With smart contract and blockchain, there is no need for a trustworthy CA, no cost on storage, and no overhead involved in key management. Figure 9 shows the smart contract structure for DPKI.

Multiple administrators can be enrolled to enhance the security of this satellite system. This can be realized by applying (n, t) threshold secret sharing or multisignature cryptography, which needs the majority of administrators to agree when taking an operation. We can also simply deploy DPKI on current public blockchain platform like Ethereum or implement DPKI on consortium blockchain constructed

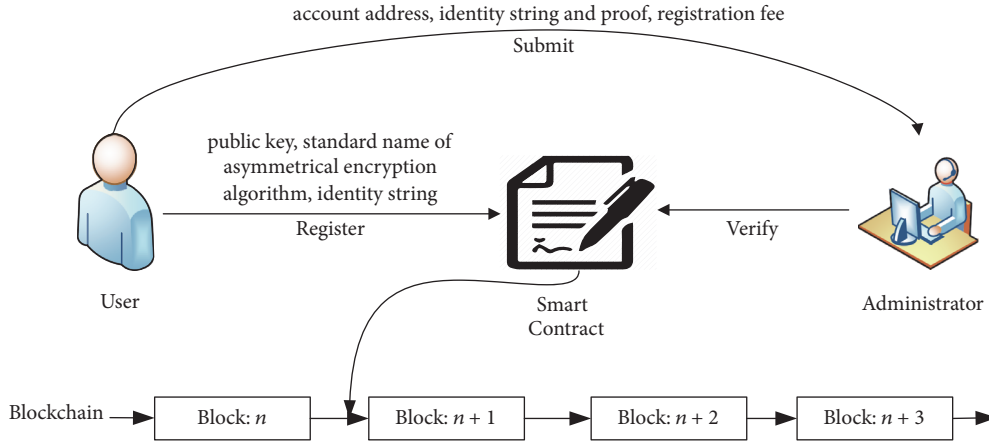


FIGURE 6: Principles of key registration.

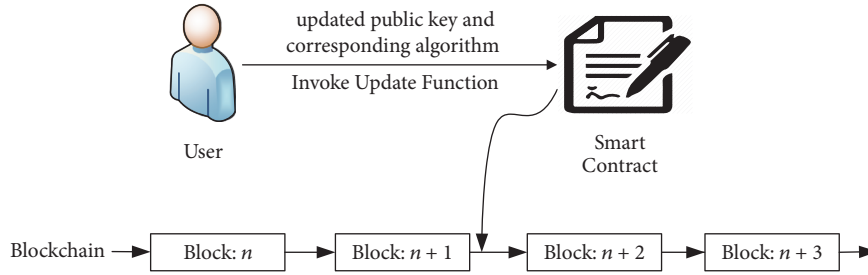


FIGURE 7: Principles of key update.

by the union of this satellite system. The two approaches with public- or consortium-based blockchain differentiates as follows:

- (1) Public-based approach does not bring any storage overhead for users or the proprietor, and there is no money cost for construction of blockchain platform.
- (2) Public-based approach must accept the current consensus mechanism by the adopted blockchain platform, while the consortium-based one can design an appropriate consensus mechanism for custom business needs, more customizable and controllable.
- (3) A constructed consortium blockchain only contains data related to key management which makes it more efficient for query and other related respects.

4.3. Analysis of DPKI. For security analysis, traditional PKI is mature and adequate while CA is completely trustworthy. As for DPKI, the algorithm adopted for the generation of users' public and private key pairs should be adequately strong. Users themselves are responsible for choosing and maintaining such safety strength. In addition, each operation that invokes DPKI smart contract costs brokerage (known as gas price in Ethereum), which has a good resistance to denial-of-service attacks. With blockchain, there is no need for a trustworthy third party which avoids the potential threat in IBE. In addition, smart contract has the characteristics of atomicity and consistency.

In respect of overhead, traditional PKI has a huge cost for key management which is also complex, and every time a user wants to communicate with someone that has legal certificates, he needs to communicate with CA to verify the validity of certificates. For IBE, if the KGC is dependable, it does not store private keys of users, all of storage overhead is for public parameters and its own public and private key. Also, the KGC can be integrated with NCC at a low price, which means there is no need for a reliable third party in this system. Thus, it can be ignored. Referring to DPKI, if it is public-block based, then there is no storage overhead for users and satellites in LEO scenario. Only two communications with any full node in blockchain are needed for query of public key before authentication or other secure communication. If it is consortium-block based, then storage overhead of those full nodes will increase with the increasing amount of data.

In summary, considering that satellites cannot be set as a full node, the communication cost of querying public key is necessary. This is fatal for any efficient authentication protocol especially in high-delay scenarios like satellite networks. In future practice, consortium blockchain is also necessary because it is more controllable and customizable. With DPKI, users can cache the public keys of satellites they commonly connect to and save the calculation of satellites' public key in the proposed authentication protocol, but this is not suitable for satellites. Thus, IBE is still the best solution provided that the KGC is totally credible. We simulated the performance of this proposed protocol using IBE on OPNET. However, it

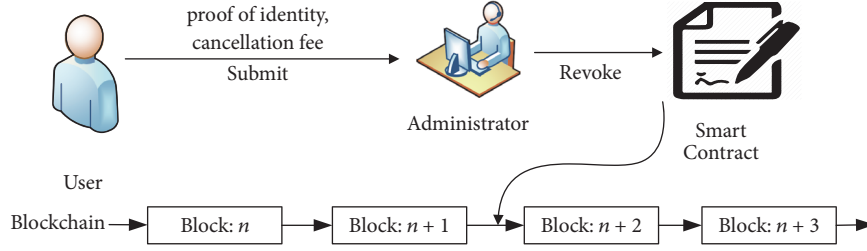


FIGURE 8: Principles of key revocation.

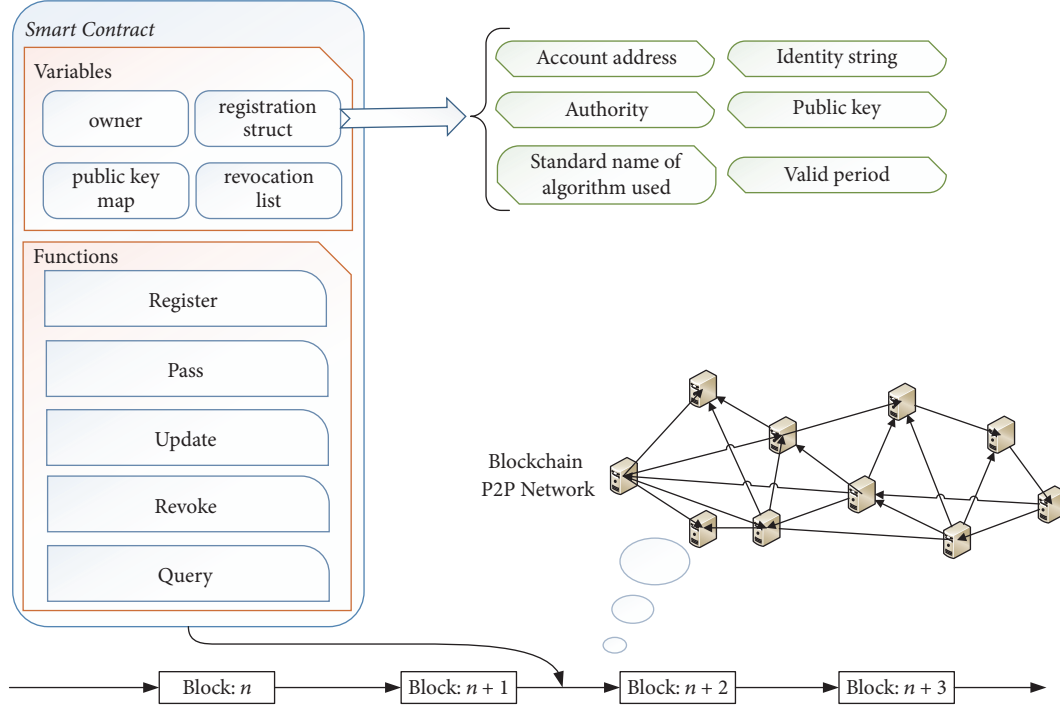


FIGURE 9: Smart contract of DPKI.

is also worth pointing out that DPKI is a promising scheme not only for satellite scenario but also for other scenarios with demands of secure communication.

5. Simulation and Evaluation

We evaluate the proposed protocol with simulation using IBE. With the OpenSSL, PBC, and GMP libraries, we implement an IBE algorithm and compare it to RSA which is recommended by the ISO as the asymmetric encryption standard. For example, in Cruickshank's paper, he uses RSA to implement the function of signature and encryption. In order to analyze the performance of the proposed protocol, we implement the protocol simulation on OPNET.

5.1. Comparison between IBE and RSA. To test whether IBE can be used in practice, we compared its performance to the RSA algorithm. While implementing IBE algorithm, we used the SHA1 algorithm that produces 160-bit digest as the hash function. We use the OpenSSL RSA algorithm.

The experimental environment used by the test program is Ubuntu 16.04 LTS with 4 GB memory and 3.30 Ghz 4 Core i5-4590 processor. After running the test program for twenty times, the computational overhead of two algorithms is shown in Figure 10.

In this experiment, the bilinear pairing used by IBE is generated by the function whose prototype is *pbk_param_init_a_gen (pbk_param_t par, int rbits, int qbits)* in PBC, where *rbits* is 160 and *qbits* is 512 by default. The average time consumed for key generation, encryption, and decryption in IBE is 7.251 ms, 1.468 ms, and 1.369 ms, respectively. In the case of RSA, the time spent for key generation, encryption, and decryption is 37.817 ms, 3.753 ms, and 4.109 ms on average. It shows that IBE is superior to RSA, and this is mainly because IBE is based on bilinear pairings while RSA is based on the difficulty of decomposing a large number. Hence, the performance of IBE can satisfy the need for practical applications on satellite networks, and some advanced LEO satellite systems such as Iridium already have their own processors onboard which are superior in performance.

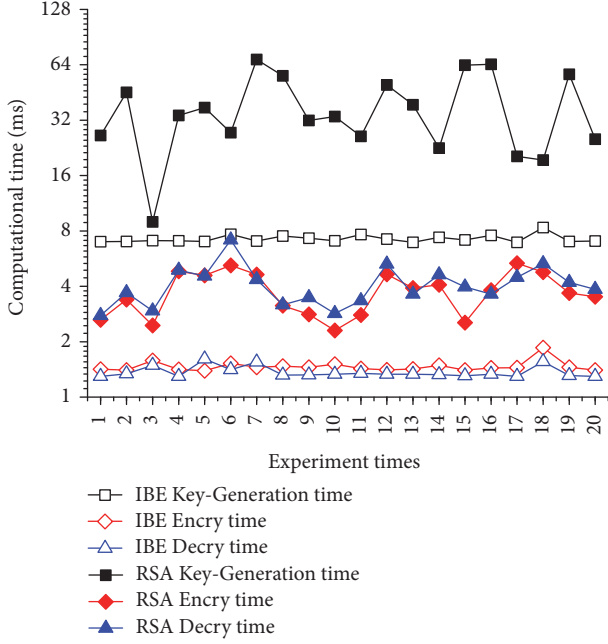


FIGURE 10: Comparison between IBE and RSA.

Moreover, the hash function, encryption, decryption, and other calculations involved in this protocol can be designed and implemented within particular hardware, so as to further reduce the demand for computing capability of satellite. In terms of the development with IBE, the Office of Chinese Security Commercial Code Administration issued the standard of SM9 algorithm which is one kind of identity-based encryption, and SM9 has entered the phase of promotion. For the security of IBE algorithm, paper [13] provides a rigorous demonstration.

5.2. OPNET Modeling and Simulation. Due to the low orbit of the satellites, handover is frequent in LEO satellite networks. Therefore, in order to ensure the communication persistence, the authentication protocol designed should be well adapted to this feature. In OPNET, we construct a LEO satellite network scenario [14] consisting of satellite nodes supporting applications attribute and analogous constellation of Iridium without backup satellites for simulation. The configurations of the satellite network include altitude: 780 km, inclination: 86.4° , period: 6027.14 s, and 6 orbits with 11 satellites per orbit.

We use *wlan_workstation_advanced* node as user node. Considering the relative motion between user and satellite, it is reasonable to set the user node to be immobile during simulation, and the satellites move in their own orbits. The process of this protocol is defined by *tasks_config*. There are mainly two phases: one is access authentication phase which is defined as *challenge_auth* and also fast-access authentication phase which is defined as *fast_access* in *task_config* object; the other one is the handover phase, which is defined as *switchsat*. The size and initialization time of message used during the simulation is based on the size of each field defined in each message and the performance of

IBE together with the symmetric encryption (using the AES-192-ECB mode). For example, random number r used in the protocol is 4 bytes, identity string is no more than 30 bytes, timestamp is 15 bytes, and separator between different fields is 2 bytes. Of course, it is just a basic simulated setting which can be adjusted according to actual business needs. The bit error rate (BER) of the intersatellite link is 10^{-4} , and the BER of mobile link between mobile user and satellite is 10^{-5} . In addition, to build the entire LEO satellite network, it is also necessary to set IP addresses, routing protocols, signal-to-noise ratio of user, satellite nodes, and so on.

5.3. Interpretation of Result. In satellite constellation scenario, we simulate the performance of the protocol in a LEO satellite network by setting custom traffic between user and satellite nodes (based on *Application config*, *Task config*, and *Profile config* object).

We first simulated a complete flow of the protocol, the whole simulation lasts for 500 s, the access authentication occurs at 150 s, the handover authentication occurs at 300 s, and the fast-access authentication occurs at 400 s. The results of simulation are shown in Table 3.

From Table 3, we can see that the response time of each phase in this protocol is less than 500 ms which is far superior to the cost of authentication in paper [15] (10 s-level) and little superior to the cost of authentication in papers [5, 16] (500 ms-level). This protocol does not affect the quality of service (QoS) of satellites with such efficient performance. At the same time, the packet delay is basically between 50 ms and 70 ms. Compared to this, the average encryption and decryption time and other processing time can be ignored, this is also the feature that a practical authentication protocol should have. Moreover, it is easy to see that the handover authentication phase saves about 100 ms to 150 ms comparing with access authentication phase, and this proves the advantages of fast handover. In addition, we can see that the response time of fast-access authentication phase is shorter than other phases which benefits from the traceability and correctness of data in blockchain, and this is far superior to the performance of authentication protocol in paper [5, 16].

Next, we adjust the simulation to make it last for two hours. During this simulation, the average interval time of handover is about 10 min which is consistent with Iridium system. The results of simulation are shown in Figures 11 and 12.

From both figures, we can see that the response time of handover authentication is about 360 ms, which is 28 percent superior to the performance of authentication protocol in paper [5, 13], and the delay is about 53 ms, which is in accordance with the result of Table 3. This also demonstrates that our proposed protocol with IBE is stable, effective, and more suitable for LEO satellite network which has the characteristic of frequent link switching.

Next, we set an additional three application scenarios and ran them for 24 hours. The configurations are listed in Table 4.

In these three application scenarios where the arrival of users conforms to the Poisson distribution, we test the

TABLE 3: Response time and delay in each phase of the protocol.

| Phase | Src | Dest | Response time | Delay |
|-------------|------|-------|---------------|-----------|
| Access | User | S_1 | 0.17771 s | 0.06737 s |
| Access | User | S_1 | 0.32246 s | 0.07591 s |
| Fast-access | User | S_2 | 0.18039 s | 0.05363 s |
| Handover | User | S_1 | 0.17816 s | 0.05322 s |
| Handover | User | S_2 | 0.20689 s | 0.05083 s |

TABLE 4: Configuration of scenarios setting.

| Application | Access (A) | Fast-access (FA) | Order |
|-------------|------------|------------------|-----------------------------|
| 1 | 10 | 10 | 10 users/1 h (concurrent) |
| 2 | 100 | 100 | 100 users/1 h (concurrent) |
| 3 | 1000 | 1000 | 1000 users/1 h (concurrent) |

TABLE 5: Simulation results of applications 1–3.

| Application | Scale | Average response time (access) | Average response time (fast-access) |
|-------------|----------------|--------------------------------|-------------------------------------|
| 1 | 10 users/1 h | 0.388472 s | 0.122618 s |
| 2 | 100 users/1 h | 0.382054 s | 0.174229 s |
| 3 | 1000 users/1 h | 0.414062 s | 0.140895 s |

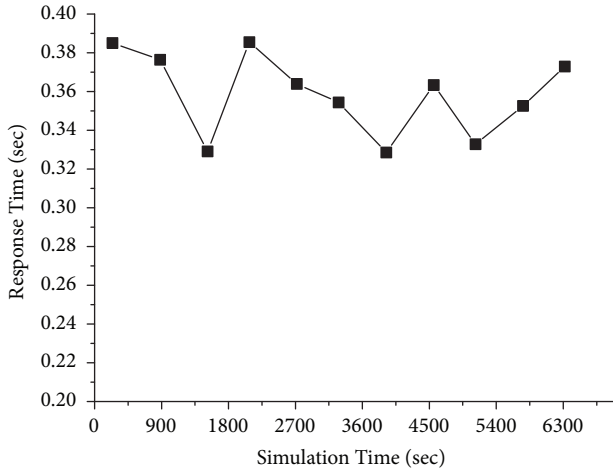


FIGURE 11: Response time.

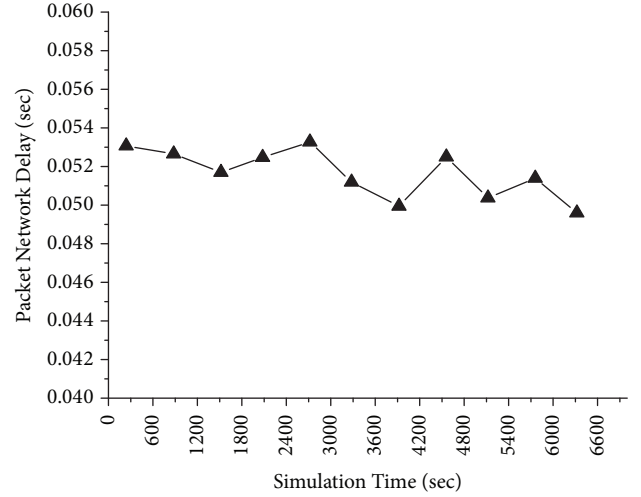


FIGURE 12: Packet delay.

efficiency and stability of this protocol with increased user-scale (10, 100, and 1000 per hour) while access and fast-access authentication phases are concurrently executed on different satellites.

From Table 5, we can find that the number of users who access the same satellite does not affect the performance of this protocol which proves this protocol stable performance. This is mainly due to the fact that there is no dependency or interference among different authentication methods. Meanwhile, the average time of these three application scenarios is about 400 ms for access authentication and 150 ms for fast-access authentication, which proves this protocol efficient performance. This is due to the low delay of LEO satellite networks compared to traditional satellite networks and the

high efficiency of IBE together with the correctness and traceability of data in blockchain. The simulation results are a little superior to the performance in Table 3 and Figures 11 and 12 which are mentioned above. This is due to the different positions of users within the satellite coverage region during authentication, which emerges when the scale of users increases. The diagram of satellite coverage region is shown in Figure 13.

Obviously, authentication response time of the user at the edge of satellite cover region would be longer than that of the user right underneath the satellite. Also, the ratio of the shortest time divided by longest time during identical authentication should be $\cos(\alpha)$ theoretically. And this explains the range of fluctuations about simulation results.

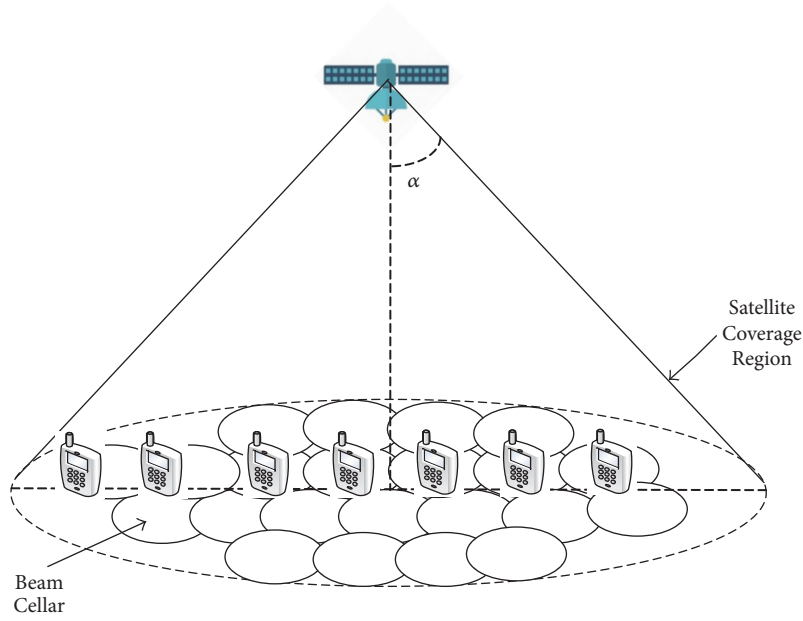


FIGURE 13: Diagram of multibeam satellite.

As for storage overhead, the cost consists of two parts: the first is used to store the public key of KGC and related system parameters; the second is for session key. Taking the number of users in Iridium system (which was 150,000 at its peak), as an example, the storage used for storing session keys is about 24 MB when 150,000 users are all online at the same time. Therefore, the cost of key storage is much lower than this for each satellite, which is acceptable. Furthermore, the logging function of this protocol also brings cost of storage, and its size is mainly determined by the threshold for storing blocks. When the number of blocks reaches the maximum, the satellite will delete all related blocks according to the certain rule. In this respect, the threshold specified is the cost of storage for each satellite (e.g., threshold can be set to 100 MB, but with the increasing number of users, it needs to be increased).

Assume that the arrival of user conforms to the Poisson distribution and the service time obeys negative exponent distribution. The computational overhead of access, fast-access, and handover authentication is R_a , R_f , and R_h , the average number of users per hour is λ , the average service time is $1/\mu$, and the average interval time of handover is t . Thus, for each satellite, the computational overhead brought by this protocol per hour is

$$x_1 \times \frac{e^{-\lambda} \lambda^{x_1}}{x_1!} \times R_a + x_2 \times \frac{e^{-\lambda} \lambda^{x_2}}{x_2!} \times R_f + \frac{e^{-\lambda} \lambda^{(x_1+x_2)}}{(x_1+x_2)!} \times (e^{-\lambda n t} - e^{-\lambda(n+1)t}) \times n \times R_h. \quad (5)$$

And x_1 represents the number of users who get authenticated by access authentication while x_2 is the number of users who get authenticated by fast-access authentication.

6. Conclusion and Future Work

Considering the dynamic topology and frequent link switching found in LEO satellite networks, this paper proposes a new decentralized access verification protocol: BAVP with IBE for authentication and blockchain for distributed computing and storage. For evaluation, we simulate this protocol in OPNET. The theoretical analysis and simulation result show that this protocol is secure, light-weighted, and efficient in LEO satellite network. Additionally, we also propose and analyze a distributed PKI scheme: DPKI which solves the problem of KGC single point-of-failure problem.

The proposed architecture and protocols will be further developed and optimized in several ways. Blockchain can ensure the stored data is accurate and tamper-resistant, but it cannot ensure data correctness and originality. That is why a third credible party is necessary. DPKI can avoid the defect of IBE where no user private key is owned by the KGC or such other centers, no matter whether these centers are reliable or not. However, the time for querying user/satellite public keys is limited by network latency which is usually high in satellite network and much longer than encryption/decryption time. Additionally, in actual deployment of blockchain on a satellite network, there are some future works like reforming blockchain technology according to particular satellite network routing algorithm and constellation needed to do. Besides, based on DPKI, many centralized application scenarios such as social applications and third-party payment can be innovated and reformed, which is also in our future research plan.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This material is based upon work supported by the China NSF Grant no. 61472189, the CASC Innovation Fund no. F2016020013, the State Key Laboratory of Air Traffic Management System and Technology no. SKLATM201703, and the Postgraduate Research & Practice Innovation Program of Jiangsu Province no. KYCX17_0369.

References

- [1] S. Li, M. Liu, and S. Wei, "A distributed authentication protocol using identity-based encryption and blockchain for LEO network," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 446–460, Springer, Cham, Switzerland, 2017.
- [2] H. S. Cruickshank, "Security system for satellite networks," in *Proceedings of the 5th International Conference on Satellite Systems for Mobile Communications and Navigation*, pp. 187–190, IET, London, UK, May 1996.
- [3] M. S. Hwang, C. C. Yang, and C. Y. Shiu, "An authentication scheme for mobile satellite communication systems," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 4, pp. 42–47, 2003.
- [4] Y. F. Chang and C. C. Chang, "An efficient authentication protocol for mobile satellite communication systems," *ACM SIGOPS Operating Systems Review*, vol. 39, no. 1, pp. 70–84, 2005.
- [5] G. Zheng, H. T. Ma, C. Cheng, and Y. C. Tu, "Design and logical analysis on the access authentication scheme for satellite mobile communication networks," *IET Information Security*, vol. 6, no. 1, pp. 6–13, 2012.
- [6] L. Qi and L. Zhi, "Authentication and access control in satellite network," in *Proceedings of the 2010 Third International Symposium on Electronic Commerce and Security (ISECS)*, pp. 17–20, IEEE, Guangzhou, China, 2010.
- [7] E.-J. Yoon, K.-Y. Yoo, J.-W. Hong, S.-Y. Yoon, D.-I. Park, and M.-J. Choi, "An efficient and secure anonymous authentication scheme for mobile satellite communication systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, no. 86, 2011.
- [8] X. Wu, A. Zhang, J. Li, W. Zhao, and Y. Liu, "A lightweight authentication and key agreement scheme for mobile satellite communication systems," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 187–204, Springer, Cham, Switzerland, 2016.
- [9] Y. Zhang, J. Chen, and B. Huang, "An improved authentication scheme for mobile satellite communication systems," *International Journal of Satellite Communications and Networking*, vol. 33, no. 2, pp. 135–146, 2015.
- [10] J. Wu, Y. Long, Q. Huang, and W. Wang, "Design and application of IBE email encryption based on Pseudo RSA certificate," in *Proceedings of the 2016 12th International Conference on Computational Intelligence and Security (CIS)*, pp. 282–286, IEEE, Wuxi, China, 2016.
- [11] D. Patel, J. Bothra, and V. Patel, "Blockchain exhumed," in *Proceedings of the Asia Security and Privacy (ISEASP)*, pp. 1–12, IEEE, Surat, India, 2017.
- [12] R. Gangishetti, M. C. Gorantla, M. L. Das, and A. Saxena, "Threshold key issuing in identity-based cryptosystems," *Computer Standards & Interfaces*, vol. 29, no. 2, pp. 260–264, 2007.
- [13] L. Chen and Z. Cheng, "Security proof of Sakai-Kasahara's identity-based encryption scheme," in *Proceedings of the IMA International Conference on Cryptography and Coding*, pp. 442–459, Springer, Berlin, Germany, 2005.
- [14] H. Long, *OPNET Modeler and Computer Network Simulation*, Xi'an University of Electronic Science and Technology Press, Xi'an, China, 2006.
- [15] Z. B. Xu and H. T. Ma, "Design and simulation of security authentication protocol for satellite network," *Computer Engineering and Applications*, vol. 42, pp. 130–132, 2007.
- [16] X. Zhang, H. Liu, Y. Lu, and F. Sun, "A novel end-to-end authentication protocol for satellite mobile communication networks," in *Foundations and Applications of Intelligent Systems*, pp. 755–766, Springer, Berlin, Germany, 2014.

Research Article

An Efficient and Privacy-Preserving Multiuser Cloud-Based LBS Query Scheme

Lu Ou ¹, Hui Yin ^{1,2}, Zheng Qin ¹, Sheng Xiao ¹, Guangyi Yang^{3,4} and Yupeng Hu¹

¹College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China

²College of Computer Engineering and Applied Mathematics, Changsha University, Changsha, Hunan 410022, China

³School of Information Science and Engineering, Central South University, Changsha, Hunan 410083, China

⁴Hunan Institute of Metrology and Test, Changsha, Hunan 410014, China

Correspondence should be addressed to Hui Yin; yhui@ccsu.edu.cn and Zheng Qin; zqin@hnu.edu.cn

Received 24 September 2017; Accepted 31 December 2017; Published 8 March 2018

Academic Editor: Zhenyu Li

Copyright © 2018 Lu Ou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Location-based services (LBSs) are increasingly popular in today's society. People reveal their location information to LBS providers to obtain personalized services such as map directions, restaurant recommendations, and taxi reservations. Usually, LBS providers offer user privacy protection statement to assure users that their private location information would not be given away. However, many LBSs run on third-party cloud infrastructures. It is challenging to guarantee user location privacy against curious cloud operators while still permitting users to query their own location information data. In this paper, we propose an efficient privacy-preserving cloud-based LBS query scheme for the multiuser setting. We encrypt LBS data and LBS queries with a hybrid encryption mechanism, which can efficiently implement privacy-preserving search over encrypted LBS data and is very suitable for the multiuser setting with secure and effective user enrollment and user revocation. This paper contains security analysis and performance experiments to demonstrate the privacy-preserving properties and efficiency of our proposed scheme.

1. Introduction

Location-based services (LBSs) are increasingly popular in today's society. It is reported that up to 150 million people have enjoyed LBSs as early as 2014 [1]. People reveal their location information to LBS providers to obtain personalized services such as map directions, restaurant recommendations, and taxi reservations.

The most common and most important service in an LBS system is a location query service. In LBS applications, a user is able to use his powerful smartphone equipped with GPS modules to obtain accurate location information anytime and anywhere by submitting a query keyword of interest (e.g., hotel) to the LBS system. Upon receiving a location query request from the user, an LBS provider rapidly returns back a target location list, in which all locations are ranked in an ascending order based on distances between the query user and these target locations.

Every coin has two sides: although the LBS greatly facilitates people's life nowadays, the user privacy disclosure

problems for LBS applications are more and more serious. The LBS providers can mine LBS users' privacy by analyzing LBS queries or recovering the spatial correlated data [2]. For example, LBS providers can easily obtain user's mobility trace or even infer user's real identity, healthy status, hobbies, and so on [3, 4]. To address the privacy challenge in the LBS system, many solutions have been proposed such as the pseudonymity technique [5], location fuzzy [6–9], and private information retrieval in the trusted third party (TTP) [5, 7]. These schemes significantly promote the further development of LBS applications.

With the rapid development of the cloud computing, more and more LBS providers are beginning to consider to outsource their location data and services to the cloud server for enjoying the numerous advantages brought by the cloud computing such as economic savings, great flexibility, quick deployment, excellent computation performance, and abundant bandwidth resources. However, the cloud server is not fully trusted usually by the LBS providers due to being

operated by the remote commercial organizations. Once the location data is outsourced to the cloud server in the plaintext form, the data security would not be guaranteed. For example, a corrupted administrator of the cloud server may sell the location data outsourced by the LBS provider to other one for obtaining illegal profit. Presently, the most effective way to protect the confidentiality of outsourced location data is to encrypt data before outsourcing it to the cloud server [10]. On the other hand, the bare user query requests also provide more opportunities for the cloud server to mine user's privacy just like a traditional LBS system. Therefore, the user requests should also be encrypted before being submitted to the cloud server. However, data encryption makes the available location query service a challenging task, since the ciphertext no longer bears the natures of numerical computation and character match in the plaintext field. Therefore, there are two essential problems that need to be solved in a cloud-based LBS application over the encrypted outsourced location data: (1) how to find out all target locations over the encrypted location data according to the encrypted user request; (2) how to compute or compare distances between these target locations and user current location over the encrypted outsourced location data.

A recent work in [11] sets out to explore the challenging issue that how to implement the cloud-based LBS system over encrypted location data and proposes a privacy-preserving cloud-based LBS query scheme, called "EPQ." The scheme enables the cloud server to perform LBS query over the encrypted LBS data without divulging users' location information. However, the scheme only can enforce a user location coordinate query according to a user's current location. In a practical LBS application, a goal-oriented keyword query is necessary for the user to accurately locate locations of interest (e.g., the user may need to accurately search hotels near to him/her). Compared with the existing work, in this paper, we propose an efficient and secure keyword-based query scheme that allows the user to be able to first accurately locate desirable locations according to the encrypted query request and then rank distances between these target locations and the user's current location, which greatly improves the user's location server experiences. Moreover, our scheme is very suitable for a multiuser cloud environment by equipping with flexible users enrollment and users revocation mechanisms.

In this paper, we make the following three key contributions:

- (i) First, we propose an efficient and privacy-preserving cloud-based LBS query scheme. For protecting the security of location data and user requests against the curious cloud server, we adopt a hybrid encryption to encrypt the outsourced location data and user requests while the cloud server can still provide accurately LBS query services for users by performing privacy-preserving and efficient search over the encrypted locations data. In addition, our scheme is very suitable for the multiuser setting by equipping with flexible user enrollment and user revocation mechanisms.
- (ii) Second, we provide detailed correction analysis and security analysis. The analyses show that our scheme is correct and can achieve user privacy preservation and confidentiality of LBS data, simultaneously.
- (iii) Lastly, we implement our scheme in Java and evaluate the performance on a real data set. Experimental results demonstrate that our proposed scheme is efficient and practical.

The rest of our paper is organized as follows. In Section 2, we review some related literatures. In Section 3, we recall a bilinear pairing map, secure kNN, and a difficulty assumption of discrete logarithm problem as the preliminaries. Then, we formalize a system model and a threat model and depict problem statements in Section 4. We present our approach in Section 5. What is more, some analyses and performance evaluations are conducted in Sections 6 and 7, respectively. Finally, we draw our conclusions in Section 8.

2. Related Work

In this section, we review some related works about privacy protection in traditional LBSs and cloud-based LBSs, respectively.

2.1. Traditional LBS Privacy Protection. Privacy leakage problem in traditional LBSs has drawn much attention of researchers, and we review mainly related literatures.

Firstly, a location k -anonymity model is introduced, which guarantees that an individual cannot be identified from $k - 1$ other individuals [12]. What is more, in a distributed environment, an anonymous approach based on homomorphic encryption [13] is proposed to protect location privacy. However, when the anonymous region is sensitive or k individuals are the same place, the sensitive location will still be leaked. Thus the third party (TTP) is proposed to manage the location information centrally [14–16]. To achieve an accurate query, a method is proposed to convert original locations of LBS data and query, maintaining a spatial relationship between the LBS data and query [16]. However, because of many users' sensitive information in the TTP, attackers would aim at attacking it easily. Then a scheme without the TTP is proposed, which protects the locations through private information retrieval [7]. Recently, considering mobile nodes, a distributed anonymizing protocol based on peer-to-peer architecture is proposed [17]. A specific zone is responded by each mobile node. Besides, an information-theoretic notion was introduced to protect privacy in LBS systems [18]. An approach is proposed to protect both client's location privacy and the server's database security by improving the oblivious transfer protocol [19]. For providing privacy-preserving map services, a new multiple mix zones location privacy protection is proposed. By using this method, users are able to query a route between two endpoints on the map, without revealing any confidential location and query information [20].

2.2. Cloud-Based LBS Privacy Protection. Considering the low computation and communication cost, the LBS providers

APSE Scheme

Key: a $(d + 1) \times (d + 1)$ invertible matrix M .

Tuple Encryption Function E_T : Consider an LBS data p which will be stored in a cloud server.

(1) Create a $(d + 1)$ -dimensional point $\hat{p} = (p^T, -0.5\|p\|^2)^T$.

(2) The encrypted data $p' = M^T \hat{p}$.

Query Encryption Function E_Q : Consider an LBS query q .

(1) Generate a random number $r > 0$.

(2) Create a $(d + 1)$ -dimensional point $\hat{q} = r(q^T, 1)^T$.

(3) The encrypted query $q' = M^{-1} \hat{q}$.

Distance Comparison Operator A_e :

Let p'_1 and p'_2 be the encrypted data of p_1 and p_2 respectively. To determine whether (p_1) is nearer to a query q than p_2 is, the system checks whether $(p'_1 - p'_2) \cdot q' > 0$, where q' is the encrypted point of q .

Decryption Function D : Consider an encrypted data p' .

The original data $p = \pi_d M^{T^{-1}} p'$, where π_d is a $d \times (d + 1)$ matrix which projects on the first d dimensions and $\pi_d = (I_d, 0)$ where I_d is the $d \times d$ identity matrix.

ALGORITHM 1: APSE scheme.

outsource the LBS data to the cloud server to compute accurate LBS queries, whereas the cloud server is semitrusted. Hence the privacy problem is still a challenge in the cloud-based LBS. There are some literatures about this problem. Firstly, a spatiotemporal predicate-based encryption is proposed for fine-grained access control [21]. Then an improved homomorphic encryption [11] is proposed to protect users' privacy and LBS data privacy. A privacy extension in crowdsourced LBS [22] is proposed. To handle the long-term privacy protection and fake path generation for LBS, a dummy-based long-term location privacy protection [23] is proposed. Recently, two-tier lightweight network coding based on pseudonym scheme in a group LBS [24] is proposed to protect privacy. What is more, a query scheme by using Bloom filter and bilinear pairing is proposed [25]. However, the literatures above did not consider the multiusers condition (i.e., joining of registered users and revocation of expired users). But unregistered users and expired users access for cloud-based LBSs is a typical scenario. Therefore, providing an efficient and privacy-preserving cloud-based LBS in multiuser environments is an unnegligible issue.

3. Preliminaries

In this section, we introduce several necessary tools used in our scheme, including a bilinear pairing map, secure kNN computation techniques, and the difficulty assumption of discrete logarithm problem.

3.1. Bilinear Pairing Map. Let \mathbb{G}_1 and \mathbb{G}_2 be two multiplication cyclic groups with large prime order q . A bilinear pairing map [26, 27] $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties:

- (i) Bilinear: for all $x, y \in \mathbb{Z}_q^*$ and $P, Q \in \mathbb{G}_1$, $e(P^x, Q^y) = e(P^y, Q^x) = e(P, Q)^{xy}$.
- (ii) Nondegenerate: if $P, Q \in \mathbb{G}_1$, then $e(P, G) \neq 1 \in \mathbb{G}_2$.
- (iii) Computable: for any element $P, Q \in \mathbb{G}_1$, there exists a polynomial time algorithm to compute $e(P, Q)$.

3.2. Secure kNN. Secure kNN [28] enables an efficient kNN computation over encrypted data points. It adopts an asymmetric scalar-product-preserving encryption (ASPE) to achieve a distance comparison between two encrypted data vectors. We synoptically introduce the principle of this technique as follows.

Definition 1 (asymmetric scalar-product-preserving encryption). Let E be an encryption function and $E(p, K)$ be an encryption of a point p under a key K . E is an ASPE if and only if E just preserves the scalar product between encrypted data points and an encrypted query points; that is,

- (1) $p_i \cdot q = E(p_i, K) \cdot E(q, K)$, where p_i is one encrypted data point and q is one encrypted query point;
- (2) $p_i \cdot p_j \neq E(p_i, K) \cdot E(p_j, K)$, where p_i and p_j are two encrypted data points.

For ease of understanding, we describe the APSE scheme in Algorithm 1. As shown in Algorithm 1, this scheme includes five parts, that is, a key, a tuple encryption function, a query encryption function, a distance comparison operator, and a decryption function.

3.3. Difficulty Assumption of Discrete Logarithm Problem. Given a multiplication group \mathbb{G} with the prime order q , g is its generator. An element x is selected from \mathbb{Z}_q^* randomly, computing $Q = g^x \in \mathbb{G}$. The definition of the difficulty assumption of discrete logarithm problem (DLP) is as follows.

Definition 2 (difficulty assumption of discrete logarithm problem). Given \mathbb{G} and g , it is difficult to compute the correct value of x . In other words, given a tuple (\mathbb{G}, q, g^x, g) , there is not an efficient polynomial time algorithm to output x .

4. Background

In this section, we formally introduce our system model and threat model and then state our proposed problem.

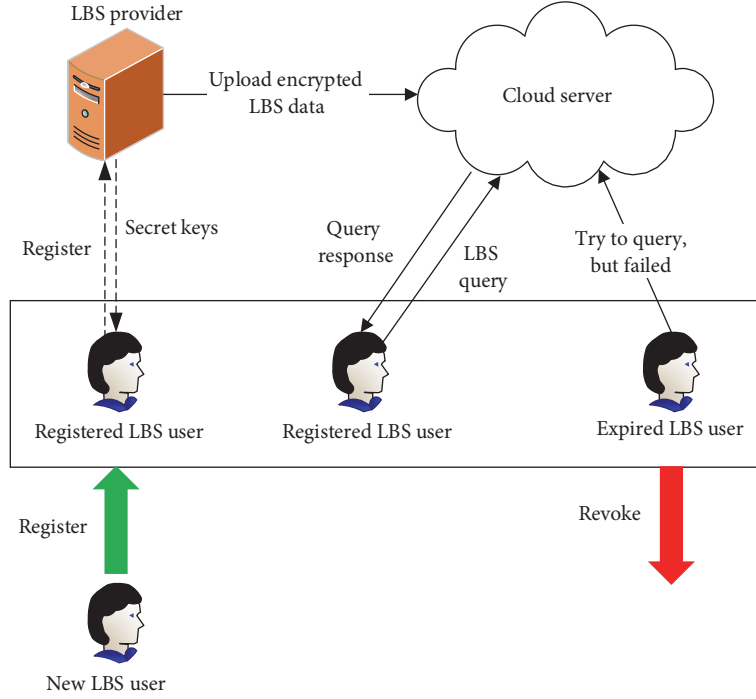


FIGURE 1: System model.

4.1. System Model. In our system model, there are three entities: an LBS provider, a cloud server, and a group of LBS users, as shown in Figure 1. Next, we introduce each entity of our model as follows.

(i) **LBS Provider.** An LBS provider is a location data owner. It outsources large-scale location data to the cloud server for enjoying the low-cost storage services and powerful computation services. To ensure the confidentiality of location data, all location data is uploaded to the cloud server after being encrypted by the LBS provider. In addition, when an LBS user wants to join the system, the LBS provider provides authentication and registration service for the LBS user. Once the LBS user passes authentication, the LBS provider sends some important security parameters to the user via secure communication channels. Correspondingly, the LBS provider is also able to revoke any expired LBS user, who no longer has the query capabilities for the outsourced location data when being revoked by the LBS provider.

(ii) **A Group of LBS Users.** A group of LBS users are the location data users, who enjoy convenient LBSs by submitting LBS query requests to the LBS provider anywhere and anytime. To hide query requests of LBS users for protecting privacy, LBS users first encrypt their query requests and then submit the encrypted query requests to the cloud server. Note that the LBS users are usually referred to the legal registered users and unregistered users and revoked users from the provider cannot enjoy LBSs.

(iii) **Cloud Server.** Upon receiving the encrypted LBS query request submitted by a legal LBS user, the cloud server

is responsible for performing the query over encrypted outsourced location data on behalf of the LBS user and returning the satisfied query results to the LBS user. In the whole query processes, the cloud server does not know any contents about outsourced location data, the user's query request, and the current location of the LBS user.

4.2. Problem Statements. In a conventional LBS system, the LBS data construction usually is organized as the category set and the location data set, as shown in Table 1(a). A *CATEGORY* denotes the general name of location data, which contains multiple concrete location data. Each concrete location data is a four-tuple $\{ID, TITLE, COORDINATE, DESCRIPTION\}$, which describes the detailed information of a certain location. When a registered user searches an interested location, he/she submits the specified *CATEGORY* and his current location coordinates to the LBS system. The LBS system first searches over the category set according to the submitted *CATEGORY* to obtain all target locations and then sorts target locations in an ascending order based on the distances between the user's current location and these target locations, which are easily computed according to the user's coordinate and each target location's coordinate, and finally returns the first k nearest locations to the query user, if the query user sends an optional parameter k to the LBS system. It means that the LBS system can analyze what are the LBS user interested in and his/her real-time location, when receiving an LBS query.

To ensure the confidentiality of LBS data and enable registered users to enforce efficient location query in the privacy-preserving manner when the LBS provider outsources LBS data and query services to the cloud server, in this paper,

TABLE 1: LBS data creation.

| (a) | |
|----------------------------|---|
| $\{CATEGORY\}$ | Location Data Set $\{ID, TITLE, COORDINATE, DESCRIPTION\}$ |
| {School: | { |
| | {1, Hunan University, (x_1, y_1) , {211, 985}}, |
| | \vdots |
| | {15, Center South University, (x_{15}, y_{15}) , {211, 985}}. |
| Hospital: | { |
| | {16, the Second Xiangya Hospital, (x_{16}, y_{16}) , {grade III-A hospital}}, |
| | \vdots |
| | {31, the Union Hospital, (x_{31}, y_{31}) , {grade III-A hospital}}. |
| \vdots | \vdots |
| Hotel: | { |
| | {450, Huatian Hotel, (x_{450}, y_{450}) , {5 stars}}, |
| | \vdots |
| | {500, Westin Hotel, (x_{500}, y_{500}) , {4 stars}}. |
| } | } |
| (b) | |
| $\{CATEGORY\}$ | Location Data Set $\{ID, TITLE, COORDINATE, DESCRIPTION\}$ |
| $\{E_1(\text{School})\}$: | { |
| | {1, $E_3(\text{Hunan University})$, $E_2((x_1, y_1))$, $E_3(\{211, 985\})$ }, |
| | \vdots |
| | {15, $E_3(\text{Center South University})$, $E_2((x_{15}, y_{15}))$, $E_3(\{211, 985\})$ }.} |
| $E_1(\text{Hospital})$: | { |
| | {16, $E_3(\text{the Xiangya Hospital})$, $E_2((x_{16}, y_{16}))$, $E_3(\{\text{grade III-A hospital}\})$ }, |
| | \vdots |
| | {31, $E_3(\text{the Union Hospital})$, $E_2((x_{31}, y_{31}))$, $E_3(\{\text{grade III-A hospital}\})$ }} |
| \vdots | \vdots |
| $E_1(\text{Hotel})$: | { |
| | {450, $E_3(\text{Huatian Hotel})$, $E_2((x_{450}, y_{450}))$, $E_3(\{5 \text{ stars}\})$ }, |
| | \vdots |
| | {500, $E_3(\text{Westin Hotel})$, $E_2((x_{500}, y_{500}))$, $E_3(\{4 \text{ stars}\})$ }.} |
| } | } |

we adopt a hybrid encryption mechanism to encrypt the LBS data; the encryption version of LBS data is shown in Table 1(b), where E_1 , E_2 , and E_3 denote different encryption scheme, respectively, whose constructions will be formally proposed in the next section. By encrypting different fields of LBS data using the different encryptions E_1 , E_2 , and E_3 , our scheme allows the cloud server to provide totally the same query service over encrypted location data as the plaintext environment aforementioned while information about the location data and user's query request is exposed to the cloud server.

From the point of view of LBS users, compared with the LBS system in the plaintext environment, the only difference

in our scheme is that an LBS user needs to encrypt the interested *GATEGORY* and his/her location coordinates to generate a query trapdoor. The cloud server performs the LBS query over encrypted outsourced location data according to the query trapdoor. Of course, the necessary decryption operations need to be involved for the LBS user once the encrypted LBS query results are received; however, this is not our concerned problem in this paper.

In addition, the LBS system is a typical multiuser application, our scheme designs efficient and flexible user registration and user revocation mechanisms to guarantee that only registered users are able to use the LBS system and unregistered users or revoked users have not access to it.

5. A Privacy-Preserving Multiuser LBS Query Scheme Based on Hybrid Encryption

In this section, we describe the implementation details of our privacy-preserving multiuser LBS query scheme. From the system-level point of view, our scheme includes six key modules: system initialization, new user grant, location data encryption, query trapdoor generation, search over encrypted location data, and user revocation. Each module is operated by one entity independently or multiple entities interactively and all modules integrally constitute our privacy-preserving multiuser LBS query system.

5.1. System Initialization. The system initialization operation is executed by the LBS provider to set up the system running environment. The LBS provider takes a large security parameter l as input and first generates two multiplication cyclic groups \mathbb{G}_1 and \mathbb{G}_2 with the large prime order q equipping the bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Let g be a generator of \mathbb{G}_1 . Then, the algorithm defines a cryptographic hash function $h_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$, which maps a message of arbitrary length to an element in \mathbb{G}_1 . Lastly, the algorithm chooses a random value $x \in \mathbb{Z}_q^*$ and generates a 3×3 invertible matrix M that are kept secretly by the LBS provider and opens the public parameter $PK = \{\mathbb{G}_1, \mathbb{G}_2, e, q, g, h_1\}$.

5.2. New User Grant. When a new LBS user u wants to join the system, the LBS provider registers the new user in this phase. At first, the LBS provider selects a random value $x_u \in \mathbb{Z}_q^*$ for u and computes g^{x/x_u} and the inverse matrix M^{-1} of M . Then, g^{x/x_u} , M^{-1} , and x_u are sent to the user u by secure communication channels. When u receives g^{x/x_u} , M^{-1} , and x_u , he/she randomly selects $r_u \in \mathbb{Z}_q^*$ and keeps r_u , M^{-1} secretly and then further computes g^{r_u} .

According to the received value g^{x/x_u} , u computes his/her register secret key $Enrk_u$:

$$Enrk_u = g^{x/x_u} \times g^{r_u} = g^{x/x_u + r_u}. \quad (1)$$

At last, $(u, Enrk_u)$ is sent to the cloud server and the cloud server stores this tuple into a user list $U-Enrkey$.

5.3. Location Data Encryption. To guarantee the security of location data, the LBS provider needs to encrypt all location information before outsourcing them to the cloud server. In this paper, to enable an efficient and privacy-preserving LBS query, we use different encryption mechanisms to encrypt the different attributes of the location data. Without loss of generality, we use $\{ID : ID_d, TITLE : T_d; COORDINATE : (x_d, y_d); DESCRIPTION : D_d\}$ to denote any location data d belongs to CATEGORY C_d in category set. The LBS provider takes the following three steps to encrypt the location data d .

First, the LBS provider uses its secret value x to code C_d as $h_1(C_d)^x$ and further employs the bilinear pairing map e to calculate $e(h_1(C_d)^x, g)$. We use E_1 to denote the code operation of CATEGORY attribute of the location data such that

$$E_1(C_d) = e(h_1(C_d)^x, g). \quad (2)$$

Second, the LBS provider uses the secretly preserved invertible matrix M to encrypt d 's coordinate (x_d, y_d) as

$$E_2(x_d, y_d) = M^T(x_d, y_d, -0.5(x_d^2 + y_d^2))^T. \quad (3)$$

Here, correspondingly, we use E_2 to denote this encryption operation.

Third, for the remaining other attributes TITLE and DESCRIPTION, the LBS provider adopts any semantically secure symmetric encryption such as AES under a given key sk to encrypt them, denoted as E_3 in our paper such that

$$\begin{aligned} E_3(T_d) &= AES_{sk}(T_d), \\ E_3(D_d) &= AES_{sk}(D_d). \end{aligned} \quad (4)$$

5.4. Query Trapdoor Generation. To preserve user's query privacy and enable correct search over encrypted location data, a query user u with the current location coordinate (x_u, y_u) needs to encrypt his/her query request before it is submitted to the cloud server. In this paper, a query trapdoor generation is conducted in two steps.

First, the user u chooses a desired query objective denoted as q (e.g., $q = Hotel$) and uses the secret value x_u granted by the LBS provider and the secret value r_u randomly chosen by himself/herself in the user grant phase to encrypt q as $h_1(q)^{x_u}$ and $h_1(q)^{x_u r_u}$.

Second, the user u generates a random number $r > 0$ and uses the matrix M^{-1} granted by the LBS provider to encrypt the current location coordinate (x_u, y_u) as $M^{-1}(rx_u, ry_u, r)^T$. Ultimately, the query trapdoor of q can be denoted as follows:

$$\mathcal{T}_u(q) = (h_1(q)^{x_u}, h_1(q)^{x_u r_u}, M^{-1}(rx_u, ry_u, r)^T). \quad (5)$$

5.5. Search over Encrypted Location Data. After the query user u generates a query trapdoor $\mathcal{T}_u(q)$, he/she submits $\mathcal{T}_u(q)$ and a parameter k to the cloud server. Upon receiving the query trapdoor $\mathcal{T}_u(q)$ and k , the powerful cloud server is responsible for searching over encrypted outsourced location data on behalf of the query use, without knowing any plaintext information of the outsourced location data and the user query request. If the user u is a legal user, the cloud server returns back the first k encrypted target locations that satisfy the query and are nearest to the query user. Therefore, in the whole query process, the cloud server must perform two key operations under the encrypted environment: (1) searching the encrypted category set according to the query trapdoor to obtain all target locations; (2) sorting the distances between target locations and user's current location in an ascending order. To achieve the above goal, the cloud sever processes the search in two steps.

First, the cloud server looks up the query user u 's registration information $\langle u, Enrk_u \rangle$ from the user list $U-Enrkey$. If the user information does not exist, the cloud server rejects the query; otherwise it linearly scans the encrypted category set and obtains all encrypted target location data if it finds out

an encrypted *CATEGORY* $E_1(C)$ in the encrypted category set that satisfies the following equation:

$$E_1(C) = \frac{e(Enrk_u, h_1(q)^{x_u})}{e(g, h_1(q)^{x_u r_u})}. \quad (6)$$

Second, upon obtaining all target locations, the cloud server sorts the distances between target locations and user's query location by evaluating

$$\begin{aligned} & \left(M^T(x_i, y_i, -0.5(x_i^2 + y_i^2))^T \right. \\ & \quad \left. - M^T(x_j, y_j, -0.5(x_j^2 + y_j^2))^T \right) \\ & \quad \cdot (M^{-1}(rx_u, ry_u, r)^T) > 0, \end{aligned} \quad (7)$$

where i and j are any two locations satisfying the query with the encrypted coordinate $E_2(x_i, y_i)$ and $E_2(x_j, y_j)$, respectively.

If the above inequality holds, then this indicates that the target location i is closer to the query user u than the target location j . Hence, i is sorted in the front of j . Finally, the cloud server returns the first k encrypted locations to the query user u .

5.6. User Revocation. User revocation is an essential yet challenging task in a practical multiuser application such as an LBS system. In some related literatures supporting user revocation, to prevent revoked users from continuing to access outsourced cloud data, the data provider usually has to rebuild data index or reencrypt large amounts of data and reuploads them to the cloud server and issues new keys to the remaining users. It unavoidably brings heavy computation and communication cost for the data provider because of the high of dynamic of users in the cloud environment. In this paper, we propose an efficient user revocation mechanism without any data reencryption and keys update operations while being able to effectively prevent the revoked user from searching outsourced location data. More concretely, for a user u' who will be revoked by the LBS provider, the LBS provider first sends the user information about u' to the cloud server. Then, the cloud server scans user information in the user list $U-Enrkey$ to find out the information of u' and deletes $(u', Enrk_{u'})$. Once $(u', Enrk_{u'})$ is deleted from $U-Enrkey$, u' no longer has the capability to search location data stored at the cloud server. Since without $Enrk_{u'}$, the cloud server cannot complete matching between the trapdoor and encrypted *CATEGORY* according to the query scheme proposed in Section 5.5, u' can still generate a legal query trapdoor.

6. Analysis

In this section, we analyze the search correctness and security to prove that our proposed scheme is correct and secure.

6.1. Search Correctness Analysis. When an authorized query user u submits his/her query trapdoor $\mathcal{T}_u(q)$ to the cloud

server, the cloud server firstly obtains the all encrypted locations satisfying the query q by performing a matching operation between an encrypted *CATEGORY* and $\mathcal{T}_u(q)$. Specifically, the cloud server judges whether $E_1(C) = e(Enrk_u, h_1(q)^{x_u})/e(g, h_1(q)^{x_u r_u})$ holds or not for an encryption *CATEGORY* $E_1(C)$. If the equation holds, then this indicates that the query q correctly matches $E_1(C)$ and the cloud server obtains all target locations belong to *CATEGORY* C . The correctness can be easily verified as follows:

$$\begin{aligned} & \frac{e(Enrk_u, h_1(q)^{x_u})}{e(g, h_1(q)^{x_u r_u})} = \frac{e(g^{x/x_u+r_u}, h_1(q)^{x_u})}{e(g, h_1(q)^{x_u r_u})} \\ & = \frac{e(g^{x/x_u}, h_1(q)^{x_u}) e(g^{r_u}, h_1(q)^{x_u})}{e(g, h_1(q)^{x_u r_u})} \\ & = \frac{e(g, h_1(q)^{x_u(x/x_u)}) e(g, h_1(q)^{x_u r_u})}{e(g, h_1(q)^{x_u r_u})} \\ & = e(h_1(q)^x, g) \\ & = e(h_1(C)^x, g) \quad (C = q) \\ & = E_1(C). \end{aligned} \quad (8)$$

Then, for any two target locations i and j , the cloud server is able to determine whether i is closer to the query current location than j by evaluating

$$\begin{aligned} & \left(M^T(x_i, y_i, -0.5(x_i^2 + y_i^2))^T \right. \\ & \quad \left. - M^T(x_j, y_j, -0.5(x_j^2 + y_j^2))^T \right) \\ & \quad \cdot (M^{-1}(rx_u, ry_u, r)^T) > 0. \end{aligned} \quad (9)$$

This is because that

$$\begin{aligned} & \left(M^T(x_i, y_i, -0.5(x_i^2 + y_i^2))^T \right. \\ & \quad \left. - M^T(x_j, y_j, -0.5(x_j^2 + y_j^2))^T \right) \\ & \quad \cdot (M^{-1}(rx_u, ry_u, r)^T) \\ & = \left(M^T(x_i, y_i, -0.5(x_i^2 + y_i^2))^T \right. \\ & \quad \left. - M^T(x_j, y_j, -0.5(x_j^2 + y_j^2))^T \right)^T \\ & \quad \cdot (M^{-1}(rx_u, ry_u, r)^T) = \left((x_i, y_i, -0.5(x_i^2 + y_i^2))^T \right. \\ & \quad \left. - (x_j, y_j, -0.5(x_j^2 + y_j^2))^T \right)^T \\ & \quad \cdot M^T M^{-1}((rx_u, ry_u, r)^T) = 0.5r((x_i^2 + y_i^2) \end{aligned}$$

$$\begin{aligned}
& - (x_i^2 + y_i^2) + 2(x_i x_u + y_i y_u - x_j x_u - y_j y_u)) \\
& = 0.5r \left(d((x_j, y_j), (x_u, y_u)) \right. \\
& \quad \left. - d((x_i, y_i), (x_u, y_u)) \right),
\end{aligned} \tag{10}$$

where $d((x_i, y_i), (x_u, y_u))$ ($d((x_j, y_j), (x_u, y_u))$, resp.) denotes the Euclidean distance between the location i (j , resp.) and the user's current location. So,

$$\begin{aligned}
& \left(M^T (x_i, y_i, -0.5(x_i^2 + y_i^2)) \right)^T \\
& - M^T (x_j, y_j, -0.5(x_j^2 + y_j^2)) \Big)^T \\
& \cdot \left(M^{-1} (rx_u, ry_u, r) \right)^T > 0 \\
& \iff 0.5r \left(d((x_j, y_j), (x_u, y_u)) \right. \\
& \quad \left. - d((x_i, y_i), (x_u, y_u)) \right) > 0 \\
& d((x_j, y_j), (x_u, y_u)) > d((x_i, y_i), (x_u, y_u)).
\end{aligned} \tag{11}$$

Thus, the cloud server is able to sort all target locations according to the above distance comparisons in an ascending order and returns back the first k nearest locations to the query user.

6.2. Security Analysis. In our proposed scheme, three encryption schemes E_1 , E_2 , and E_3 are employed to protect the confidentiality of LBS data. In this section, we will analyze the security of our scheme against the “honest-but-curious” cloud server in the multiuser environment.

E_2 is a semantically secure symmetric encryption such as AES that encrypts the TITLE and DESCRIPTION fields of LBS data. The semantic security of AES guarantees the security of TITLE and DESCRIPTION fields of LBS data. We use secure kNN encryption technique denoted as E_2 to encrypt the COORDINATE attribute of LBS data and query user's coordinate to enable a secure and effective distance comparison. The security of the COORDINATE attribute of LBS data and the query user's coordinate mainly relies on the security of secure kNN scheme. For the CATEGORY attribute of LBS data, we use E_3 to encrypt it to enable secure and flexible search over encrypted location data. Specifically, given a location data with CATEGORY attribute C , the ciphertext can be denoted as $E_3(C) = e(h_1(C)^x, g) = e(h_1(C), g)^x$. Since $e(h_1(C), g)$ is a group element in \mathbb{G}_2 with a large prime order, the secret key x is acknowledgedly intractable from $E_3(C)$ in the large number field due to the well-known DLP assumption. Without the secret key x kept secretly by the LBS provider, the cloud server cannot recover the message C from encryption E_3 .

In addition, in the multiuser environment, the system should prevent an illegal user from requesting for query LBS data stored in the cloud server. In our scheme, when a registered user u wants to query the LBS location data using q , u uses secret query keys x_u and r_u to generate the

query trapdoor of q , $trap_u(q) = (h_1(q)^{x_u}, h_1(q)^{x_u r_u})$. Under the assumption of DLP, attackers cannot compute out x_u and r_u according to $trap_u(q)$. Without the correct x_u and r_u , an illegal user u_l cannot generate the correct query trapdoor such that u_l cannot perform the correct query over encrypted location data. For a revoked user u_R , although u_R can generate the trapdoor $trap_{u_R}(q)$ for q , u_R still cannot let the cloud server perform a correct query on behalf of him/her due to lacking of the necessary query parameter $Enrk_{u_R}$ that has been deleted from the list $U-Enrkey$ by the cloud server in the phase of user revocation.

7. Evaluation

In this section, we evaluate the performance of our proposed scheme from the perspective of the LBS provider, the LBS user, and the cloud server, respectively. The software and hardware configurations of the LBS provider and LBS user side are Windows 7 desktop systems with Intel Core 2 Duo CPU 2.26 GHZ, 3 GB memory, and 320 GHZ hard driver and the cloud sever side is a virtual machine with Core 2 Duo CPU 4×2.394 GHZ, 8 GB memory on the Dell blade sever M610, and the Linux Centos 5.8 OS.

All programs are developed using Java language and the JPBC library [29] is employed to implement group operations. We execute all experiments in a real data set collected from the *open street map* [30] with 50 categories and the number of concrete location data being 1000 by extracting the location data belonging to Yuelu District, Changsha, China.

7.1. LBS Data Encryption. Figure 2(a) shows that the time cost of encrypting LBS data for the LBS provider increases linearly with the increasing size of category set when the total number of location data remains unchanged ($n = 1000$). Figure 2(b) shows the number of concrete location data has little influence on the time cost of encrypting LBS data when fixing the size of category set ($c = 50$). Recall that, in our scheme, E_1 is used to encrypt categories in category set and E_2 and E_3 are used to encrypt location data. Experimental results from Figure 2 illustrate that the time cost of encrypting LBS data is closely related to the encryption E_1 while not being almost affected by E_2 and E_3 . It is reasonable that the E_1 consists of the time-consuming pair operation and exponent operation over the ellipse curve group while E_2 and E_3 almost do not consume time when an extremely small message and 3-dimensional vector are encrypted by them, respectively.

7.2. Query Request Encryption. According to the query trapdoor generation proposed in the Section 5.4, in the whole processes of the query request encryption, three key operations are involved to encrypt an interested query keyword and current location coordinate for a registered LBS user (i.e., the hash operation, the exponentiation operation on group, and the matrix multiplication operation between a 3×3 matrix and a 3-dimensional column vector). The time cost of each operation based on our software/hardware setting is shown in Table 2. Therefore, the total time cost of generating a query

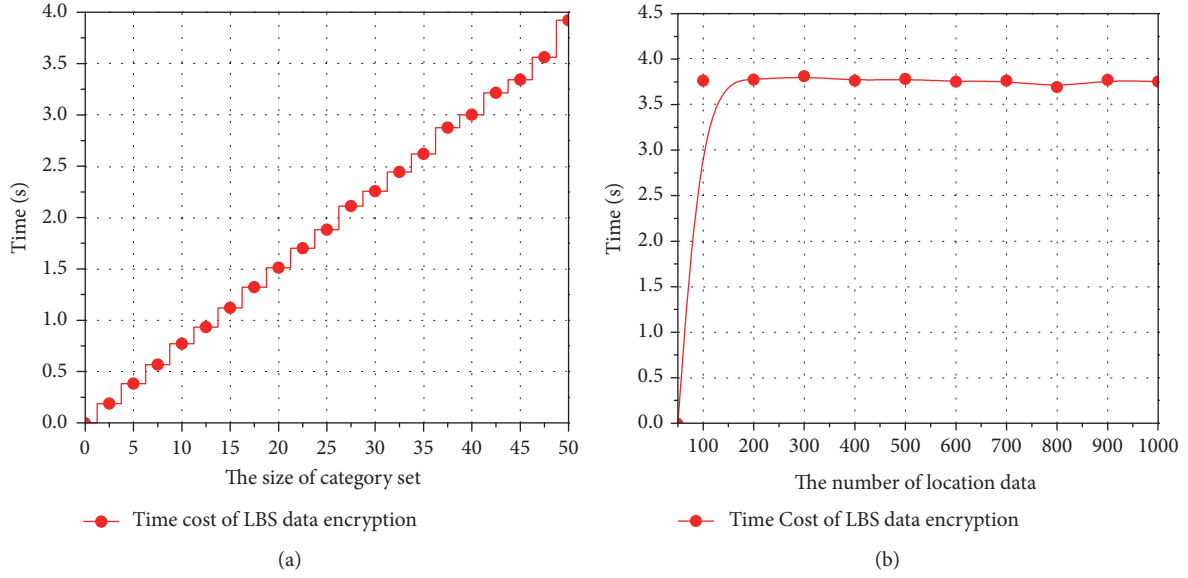


FIGURE 2: (a) The time cost of encrypting LBS data for LBS provider for different size of category set with fixed number of location data, $n = 1000$; (b) the time cost of encrypting location data for LBS provider for different number of location data with fixed size of category set, $c = 50$.

TABLE 2: Time cost of operation.

| Notations | Descriptions | Time (ms) |
|--------------------|---|--------------|
| h_1 | Hash function $h_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$ | ≈ 93 |
| $E_{\mathbb{G}_1}$ | Exponentiation operation on group \mathbb{G}_1 | ≈ 34 |
| M | Matrix multiplication operation | < 1 |

request for an LBS user can be denoted as $h_1 + 2 * E_{\mathbb{G}_1} + M \approx 161$ ms; it is extremely efficient in practice.

7.3. Query over Encrypted LBS Data. Figures 3(a) and 3(b) show the time cost of search over encrypted location data for the cloud server. We can observe that the number of categories and encrypted location data have little influence on the overhead of search for the cloud server. This is because that the main time cost for the search is to only enforce two relatively time-consuming pairing operations while linear search over 50 categories according to the query trapdoor for target location data and 3-dimensional vector computation for distance comparison almost does not consume time.

Figure 3(c) shows the average response time of our query scheme for different sizes of query users. We can see that the response time grows linearly with the increasing number of query users. When the number of query users achieves 100, the response time is about 6.82 s, and this is extremely efficient in practical application.

8. Conclusion

In this paper, we propose a privacy-preserving multiuser LBS query scheme based on the hybrid encryption in the cloud

environment. Adopting different encryptions on different attributes of LBS data, our proposed scheme can achieve users' location privacy protection and the confidentiality of LBS data. In particular, the LBS query is performed in the cipher environment, thus the LBS users can get the accurate LBS query results without disclosing their private information. Besides, we consider LBS user accountability and LBS user dynamics, for preventing the unregistered users and expired users accessing. And extensive experiments show that our proposed scheme is highly efficient. In the future, we will consider collusion attacks in the cloud-based LBSs.

Conflicts of Interest

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

Authors' Contributions

Hui Yin and Zheng Qin equally contributed to this work.

Acknowledgments

This work is partially supported by the National Science Foundation of China under Grants nos. 61472131 and 61772191; the Science and Technology Key Projects of Hunan Province under Grants nos. 2015TP1004, 2015SK2087, 2015JC1001, and 2016JC2012; the Natural Science Foundation of Hunan Province under Grant no. 2017JJ2292; Outstanding Youth Research Project of Provincial Education Department of Hunan under Grant no. 17B030; the Science and Technology Planning Project of Changsha under Grant no. k1705018.

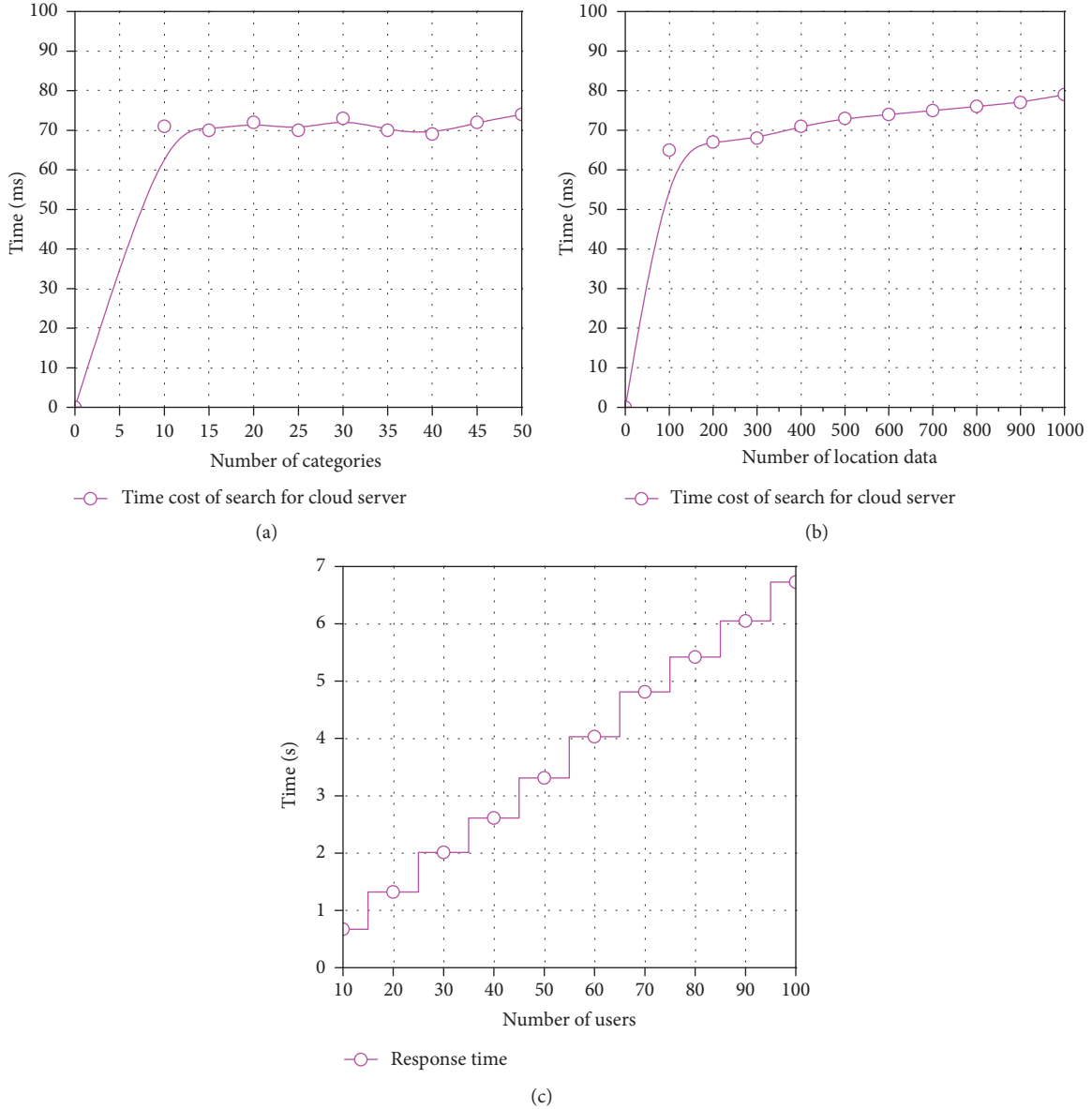


FIGURE 3: (a) The time cost of search for cloud server for different number of categories with fixed number of location data, $n = 1000$; (b) the time cost for search for cloud server for different number of location data with fixed number of categories; (c) the system response time for different number of search users with fixed number of categories and location data, $n = 1000$, $c = 50$.

References

- [1] Statista, "Number of location-based service users in the United States from 2013 to 2018 (in millions)," Statista; 2017. <https://www.statista.com/statistics/436071/location-based-service-users-usa/>.
- [2] K. Xie, X. Ning, X. Wang et al., "Recover corrupted data in sensor networks: a matrix completion solution," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1434–1448, 2017.
- [3] M. Li, H. Zhu, Z. Gao et al., "All your Location are belong to us: breaking mobile social networks for automated user location tracking," in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2014*, pp. 43–52, USA, August 2014.
- [4] J. Shao, R. Lu, and X. Lin, "FINE: a fine-grained privacy-preserving location-based service framework for mobile devices," in *Proceedings of the IEEE INFOCOM*, pp. 244–252, IEEE, Ontario, Canada, May 2014.
- [5] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 121–132, ACM, 2008.
- [6] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.
- [7] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1200–1210, 2014.

- [8] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.
- [9] Q. Wang, C. Xu, and M. Sun, "Multi-dimensional K-anonymity based on mapping for protecting privacy," *Journal of Software*, vol. 6, no. 10, pp. 1937–1944, 2011.
- [10] K. Xie, X. Ning, X. Wang et al., "An efficient privacy-preserving compressive data gathering scheme in WSNs," *Information Sciences*, vol. 390, pp. 82–94, 2017.
- [11] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7729–7739, 2016.
- [12] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pp. 31–42, ACM, San Francisco, Calif, USA, May 2003.
- [13] G. Zhong and U. Hengartner, "Toward a distributed k-anonymity protocol for location privacy," in *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society (WPES '08)*, pp. 33–37, ACM, Alexandria, VA, USA, October 2008.
- [14] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper: query processing for location services without compromising privacy," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 4, Article ID 24, 2009.
- [15] B. Gedik and L. Liu, "Location privacy in mobile systems: a personalized anonymization model," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pp. 620–629, IEEE, June 2005.
- [16] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Proceedings of the Advances in Spatial and Temporal Databases*, pp. 239–257, Springer, 2007.
- [17] M. Ghaffari, N. Ghadiri, M. H. Manshaei, and M. S. Lahijani, "P⁴QS: a peer-to-peer privacy preserving query service for location-based mobile applications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9458–9469, 2017.
- [18] Z. Montazeri, A. Houmansadr, and H. Pishro-Nik, "Achieving perfect location privacy in wireless devices using anonymization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2683–2698, 2017.
- [19] H. Jannati and B. Bahrak, "An oblivious transfer protocol based on elgamal encryption for preserving location privacy," *Wireless Personal Communications*, vol. 97, no. 2, pp. 1–11, 2017.
- [20] I. Memon, Q. A. Arain, M. H. Memon, F. A. Mangi, and R. Akhtar, "Search me if you can: multiple mix zones with location privacy protection for mapping services," *International Journal of Communication Systems*, vol. 30, no. 16, Article ID e3312, 2017.
- [21] Y. Zhu, D. Ma, D. Huang, and C. Hu, "Enabling secure location-based services in mobile cloud computing," in *Proceedings of the 2013 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing, MCC 2013*, pp. 27–32, China, August 2013.
- [22] J. B. Abdo, T. Bourgeau, J. Demerjian, and H. Chaouchi, "Extended privacy in crowdsourced location-based services using mobile cloud computing," *Mobile Information Systems*, vol. 2016, Article ID 7867206, 13 pages, 2016.
- [23] F. Tang, J. Li, I. You, and M. Guo, "Long-term location privacy protection for location-based services in mobile cloud computing," *Soft Computing*, vol. 20, no. 5, pp. 1735–1747, 2016.
- [24] J. Y. Chen and C. L. Wang, "Privacy protection for mobile cloud data: a network coding approach," <https://arxiv.org/abs/1701.07075>.
- [25] H. Yin, Z. Qin, L. Ou, and K. Li, "A query privacy-enhanced and secure search scheme over encrypted data in cloud computing," *Journal of Computer and System Sciences*, vol. 90, pp. 14–27, 2017.
- [26] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.
- [27] H. Yin, Z. Qin, J. Zhang, and K. Li, "Achieving secure, universal, and fine-grained query results verification for secure search scheme over encrypted cloud data," *IEEE Transactions on Cloud Computing*, no. 99, 2017.
- [28] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proceedings of the International Conference on Management of Data and 28th Symposium on Principles of Database Systems (SIGMOD-PODS '09)*, pp. 139–152, Providence, RI, USA, July 2009.
- [29] A. de Caro and V. Iovino, "jPBC: java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, pp. 850–855, July 2011.
- [30] Openstreetmap Foundation UK West Midlands, Openstreetmap. Openstreetmap Foundation West Midlands, U, K; 2017, <http://www.openstreetmap.org/#map=10/1.3375/103.9732>.

Research Article

Quantum Cryptography for the Future Internet and the Security Analysis

Tianqi Zhou,¹ Jian Shen ,^{1,2} Xiong Li,³ Chen Wang,¹ and Jun Shen¹

¹Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, China

²State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

³Hunan University of Science and Technology, Xiangtan, China

Correspondence should be addressed to Jian Shen; s_shenjian@126.com

Received 28 December 2017; Accepted 29 January 2018; Published 21 February 2018

Academic Editor: Guojun Wang

Copyright © 2018 Tianqi Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cyberspace has become the most popular carrier of information exchange in every corner of our life, which is beneficial for our life in almost all aspects. With the continuous development of science and technology, especially the quantum computer, cyberspace security has become the most critical problem for the Internet in near future. In this paper, we focus on analyzing characteristics of the quantum cryptography and exploring the advantages of it in the future Internet. It is worth noting that we analyze the quantum key distribution (QKD) protocol in the noise-free channel. Moreover, in order to simulate real situations in the future Internet, we also search the QKD protocol in the noisy channel. The results reflect the unconditional security of quantum cryptography theoretically, which is suitable for the Internet as ever-increasing challenges are inevitable in the future.

1. Introduction

With the popularization and rapid development of the Internet, human society has entered the information age. Nowadays, all walks of people and all aspects of life can not be separated from the network. In 1990s, the term “cyberspace” was used to represent many new ideas and phenomena in the Internet, networking, and digital communication [1]. Nowadays, this term is used to describe the domain of the global technology environment by experts and researchers of technical strategy, security, government, military, and industry and enterprises. Also, this term is used to refer to anything associated with the Internet. Using this global network, people can engage in all kinds of activities such as communicating ideas, sharing information, providing social support, conducting business, directing actions, creating artistic media, playing games, and engaging in political discussion. Typical applications based on cyberspace include cloud computing [2, 3] and personalized recommender systems [4].

Despite all benefits and advantages of cyberspace, it is regarded as the largest unregulated and uncontrolled field

in human history. Therefore, the problem of information security is the primary problem of cyberspace. On the one hand, information technology and industry have entered an unprecedented stage of prosperity. On the other hand, the means of all kinds of attacks emerge in an endless stream. Attacks, like hacker attacks, malicious software invade, and computer viruses, pose a great threat to cyberspace information security. Moreover, the progress of science and technology also poses new challenges to cyberspace security.

Due to the characteristics of the quantum computer, many existing public key cryptography (RSA [5, 6], ELGamal [7], elliptic curve cryptography (ECC) [8], and so on) will be no longer safe in the quantum computer. Namely, the well-known discrete logarithm problem (DLP) or the integer factorization problem will no longer be difficult under quantum computer. This suggests that in order to resist quantum computers, new cryptosystems that are not based on discrete logarithms problem or the large factor decomposition problem should be explored. Only in this way can the information security of cyberspace be ensured in the future Internet.

Taking protective measures at all levels and scope of the network is the basic idea of cyberspace security [9]. These measures aim at detecting and discovering all kinds of network security threats and taking corresponding response actions.

Quantum cryptography is still in its infancy. But we can not ignore the challenges it brings to the security of existing cyberspace. In 1994, mathematician Shor has proposed the quantum algorithm [10] by which the integer factorization problem and the discrete logarithm problem can be efficiently solved in polynomial time. Note that so far researchers have not found the classical algorithm to solve the large integer decomposition and the discrete logarithm problem efficiently under the Turing machine model. Therefore, the challenge of the emergence of quantum computers to the traditional cryptosystems can not be ignored even if it is still in its infancy.

Cryptography and network security are the key technologies to ensure the security of the information system [11]. Quantum cryptography is an important branch of cryptography, which is the combination of quantum mechanics and classical cryptography. The security of communication can be guaranteed by Heisenberg's uncertainty principle and quantum no-cloning theory [12]. The main goal of the study of quantum cryptography is to design cryptographic algorithms and protocols, which is against quantum computing attacks.

As stated previously, exploring quantum cryptographic protocols will be an essential part of cyberspace security issues for future Internet. In this paper, we concentrate on analyzing and exploring the quantum key distribution protocol target for cyberspace security for the future Internet.

1.1. Organization. The rest of this paper is organized as follows. Section 2 introduces some related works about quantum cryptography. Section 3 presents preliminaries of quantum physics and quantum communication. Section 4 presents benefits that quantum cryptography brings to the future Internet and analyze the security of it. Section 5 concludes our paper.

2. Relate Works

Quantum cryptography stems from the concept of quantum money, which was proposed by Wiesner in 1969. Limited by the level of technology in history, this novel and creative idea cannot be realized, which makes it remain unpublished until 1983 [13].

The first practical QKD protocol [14] was proposed by Bennett and Brassard, in 2011. By leveraging single photon polarization, they pioneered the implementation of the quantum key distribution protocol. After that, a lot of effort was put into QKD in order to improve security and efficiency. In 1991, Ekert proposed the protocol [15] that is based on Bells theorem. Note that [15] employs a pair of quantum bits (i.e., an EPR pair), which is essentially the same as [14]. Subsequently, in 1992, the improvement [16] of the scheme [14] was put forward by Bennett. Employing any two nonorthogonal states, the improvement is more efficient and simple. After that, many QKD protocols [17, 18] using the

basic principles of quantum mechanics have been proposed successively.

As an important cryptographic basic protocol, the oblivious transfer protocol is one of the key technologies for privacy protection in cryptography [19]. The oblivious transfer protocol is a protocol, where the sender sends many potential information to the receiver, but the sender itself is not aware of the specific content of the transmission. The concept of quantum oblivious transfer (QOT) [20] was first put forward by Crépeau in 1994. After that, many works have been devoted to the QOT protocol. In 1994, the "oblivious transfer" security of [21] against any individual measurement allowed by quantum mechanics was proved by Mayers and Salvail in [22]. In 1998, the protocol [23] was proposed, which proves the security of the QOT protocol under an eavesdropper. Other protocols [24, 25] were proposed to improve QOT protocol to varying degrees.

Quantum authentication (QA) protocol is also one of the quantum cryptographic protocols. It was proposed in [26] in 2001. After that, many QA protocols [27, 28] have been proposed one after another.

The quantum cryptography protocol has developed many branches now. In addition to the protocols (i.e., QKD protocol, QOT protocol, and QA protocol) we discussed above, quantum cryptography protocols also include quantum bit commitment (QBC) protocols [29, 30] and quantum signature (QS) protocols [31, 32].

3. Preliminaries

In many respects, quantum communication and information processing are superior to that of classical, which is rooted in the characteristics of quantum information.

3.1. Properties of Quantum Information. Properties of quantum information mainly include uncertainty principle, quantum no-cloning theory, the quantum teleportation, and the hidden characteristics of quantum information, which can be employed to resist attack (passive or active attack [33]) in cyberspace communication.

Heisenberg's uncertainty principle and quantum no-cloning theory [12].

- (i) Uncertainty principle: it is known as Heisenberg's uncertainty principle, which was introduced in 1927 by the German physicist Heisenberg [34]. The main idea of uncertainty principle is that the particle position in the micro world is impossible to be determined, and it always exists in different places with different probability.
- (ii) Quantum no-cloning theory [12]: quantum no-cloning theory is the uncloned and undeleting properties of the unknown quantum state. Cloning means producing a completely identical quantum state in another system. Scientists have proved that machines capable of replicating quantum systems do not exist [35]. The undeleting principle can guarantee that any deleting and damaging effect of the enemy on the quantum information will leave a trace in secure

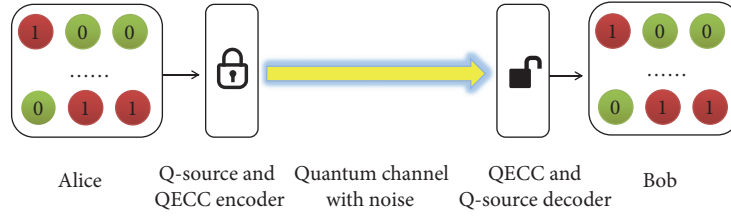


FIGURE 1: Quantum direct communication model.

communication. It was proposed in [36] in Nature that deleting a copy of an arbitrary quantum state is not allowed by linearity of quantum theory.

- (iii) Quantum teleportation: the classic information is obtained by the sender measuring the quantum state of the original, which will be told by the sender in the way of classical communication. Quantum information is the rest of the information that the sender does not extract in the measurement, and it is passed to the recipient by measurement. In 1993, the scheme that teleports an unknown quantum state was proposed in [37].
- (iv) Hidden characteristics of quantum information: quantum information has unique properties that classical information does not possess. Specifically, the information of the quantum code in the entangled state can not be obtained by the local measurement operation, which can only be revealed by joint measurement. The works about quantum information concealment was proposed in 2001 by Terhal et al. in [38].

3.2. Quantum Communication System. The quantum communication can be divided into quantum direct communication and quantum teleportation communication. Quantum direct transmission model is the simplest mode to realize the transmission of quantum signals in different places. Figure 1 depicts quantum direct communication model.

In this Figure 1, Alice wants to communicate with Bob through a quantum channel. In the quantum direct transmission model, Alice first needs to produce a series of photons through the preparation device according to the message she wants to share with Bob. After the generation of the quantum source, this information also needs to be processed by quantum source encoder and quantum error correcting code (QECC) encoder. Then, the quantum information can be transmitted directly to the quantum channel (optical fiber or atmosphere). Here, the quantum channel is easily disturbed by external noise. Therefore, the receiver Bob first performs QECC encoding to the received signal and then performs quantum source encoding. Finally, Bob obtains the initial quantum message.

The other quantum communication is the quantum teleportation. Unlike the classical communication, the qubits not only can be in a variety of orthogonal superposition states but also can be in the entangled state. The principle of quantum teleportation is to establish a quantum channel

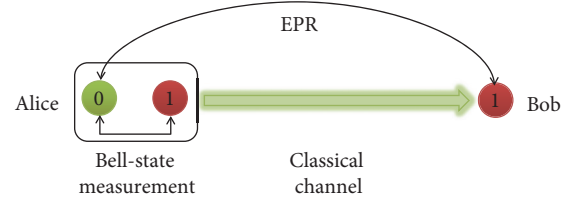


FIGURE 2: Quantum teleportation.

by using the maximum entangled state of two particles. Then the message is transmitted by the quantum operation. Note that selection of communication channels is the difference between the teleportation and the direct communication. Model of the quantum teleportation is illustrated in Figure 2.

In this model, we depict that Alice who wants to transmit one-bit quantum whit Bob in other place. Firstly, an EPR pair is generated by the EPR entanglement source. Secondly, one of the particles is sent to Alice and the other is sent to the receiver Bob through the quantum channel. Thirdly, in order to transmit information, Alice needs to measure the particles in the EPR entangled pairs and the pending bits she holds. And then, Alice informs Bob of measurement results. Finally, based on the measurement results of Alice and the measurement of the EPR pair of himself, Bob can obtain information about the particles to be transmitted.

4. Quantum Cryptography for Future Internet

Security for cyberspace in the future Internet should be guaranteed as it is the collection of all information systems and the information environment for human survival. For the growing security problem in cyberspace, quantum cryptography becomes the first consideration.

4.1. Unconditional Security. Cable and light are the main carriers of today's Internet communication. This communication system model is shown in Figure 3. Alice and Bob are legitimate users in the system while Eve is an eavesdropper. In order to ensure security, they encrypt messages and then transmit it on the public channel. The classical cryptosystem is roughly divided into two kinds, which are symmetric key cryptosystems and asymmetric key cryptosystems. For these two cryptosystems, their security is mostly based on the complexity of computing. However, the rapid development of hardware equipment and the proposed new advanced algorithms have brought unprecedented challenges to the

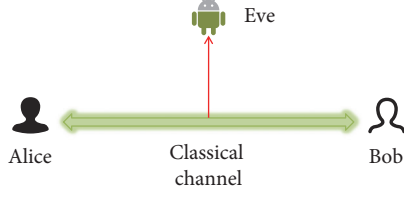


FIGURE 3: Classical communication model.

security of classical cryptosystems. Moreover, the rapid development of quantum computing has also made many difficult problems in classical mathematics have the solvability in the field of quantum physics. For example, the DLP and the integer factorization problem have been solved in [10] in 1994. Therefore, exploring quantum cryptographic protocols will be an essential part of cyberspace security issues for future Internet.

Shannon, the founder of the information theory, made a pioneering study of unconditional security in the 50s of last century [39]. In this study, unconditional security conditions of “one-time-pad” were given. Namely, rather than the pseudo-random number, the encryption/decryption key is real random. And this key is used only once. Furthermore, the key length is equal to the plaintext and performs the exclusive or operation with the plaintext by bit. However, the problem of key distribution at one-time pad has never been solved. It is worth noting that this problem of key distribution can be solved by the principle of quantum mechanics.

Figure 4 illustrates the model of the famous QKD protocol [14].

In this model, sender wants to share a common conference key with his/her counterpart, which can be used to encrypt/decrypt messages they communicate. In this QKD protocol, the real randomness of the key is guaranteed by the essential properties of the quantum: uncertainty principle. Moreover, an attacker is definitely detected if it exists.

4.2. Sniffing Detection. In Figure 3, Alice and Bob exchange information in public channel. In order to ensure confidentiality, their information is encrypted, but they cannot prevent an attacker from eavesdropping on the channel. Moreover, because of the characteristics of the device itself, the eavesdropper can not be detected whether it is in cable communications or in optical fiber communications. In cable communications, the listener can use a multimeter or oscilloscope to monitor. In optical fiber communications, the eavesdropper can get information from a part of the light signal. Note that the fiber loss is influenced by environmental factors, such as temperature and pressure, which makes the loss caused by eavesdropping not be perceived.

In quantum communication, the eavesdropper is sure to be detected owing to quantum no-cloning theory. Specifically, in Figure 4, if an eavesdropper monitors the quantum channel, for a bit of quantum information, he will choose the same measuring base with the sender with a 50% probability. Therefore, the eavesdropper will be detected at a 50% probability for a bit of quantum information. Note that,

TABLE 1: Measurement results.

| Results | Polarization | |
|-------------------|----------------|----------------|
| | \oplus | \otimes |
| Bases | | |
| \leftrightarrow | 1 | 0: 50%; 1: 50% |
| \updownarrow | 0 | 0: 50%; 1: 50% |
| \nearrow | 0: 50%; 1: 50% | 0 |
| \nwarrow | 0: 50%; 1: 50% | 1 |

for the quantum information of n -bit, the probability of the eavesdropper being detected is $1 - (1/2)^n$.

4.3. Security of the QKD. In this subsection, in order to simulate real situations in the future Internet, we first analyze the quantum key distribution protocol in noise-free channel. Moreover, we further search the quantum key distribution protocol in noisy channel.

In order to analysis security of QKD protocol, we list the encoding of quantum information and the measurement results under different measurement bases in Table 1. The two parties agree in advance that the horizontal and oblique downwards polarization represents “1” while the vertical and oblique upward polarization represents “0.”

The probability of the existence of a eavesdropper on the QKD protocol is as follows.

$$\Pr = \Pr \{ \text{Base}_A = \text{Base}_B \wedge \text{Measure}_A \neq \text{Measure}_B \}. \quad (1)$$

The probability that the eavesdropper is found for 1-bit quantum information is calculated as $1/2 \times 1/2 \times 1/2 = 1/8$.

Figure 5 illustrates the probability of the eavesdropper being detected in noise-free channel. From the graph we can see that when the number of transmissions exceeds 40, the probabilities of the eavesdropper are close to 100%. While Figure 6 illustrates the probability of the eavesdropper being detected in the channel with 30% noise. The graph shows that when the number of transmitted photons is close to 80, the probability of the eavesdropper being detected is close to 100%. From the above two figures we can conclude that the eavesdropping behavior in quantum communication is certain to be detected. In particular, the more the number of transmission data the higher the probability of the eavesdropper being detected, no matter whether there is noise interference or not.

Figure 7 shows the probability of error in the receiver when the eavesdropper eavesdrops on the channel in different probability. It indicates that the error rate of the receiver is 25% in the absence of eavesdropper, while that of the receiver is about 31% when the eavesdropper monitors the channel with a probability of 50% and that of the receiver rises to about 37% when the eavesdropper monitors every bit of the channel.

Figure 8 shows the eavesdropper being detected when he/she eavesdrops on the channel in different probability. In this picture, the purple line represents that the attacker monitors the channel in the possibility of 100% while the green line and the red line represent that the attacker

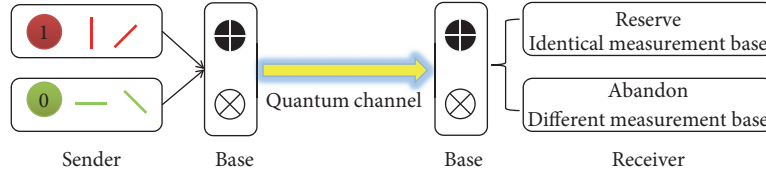


FIGURE 4: Model of QKD protocol.

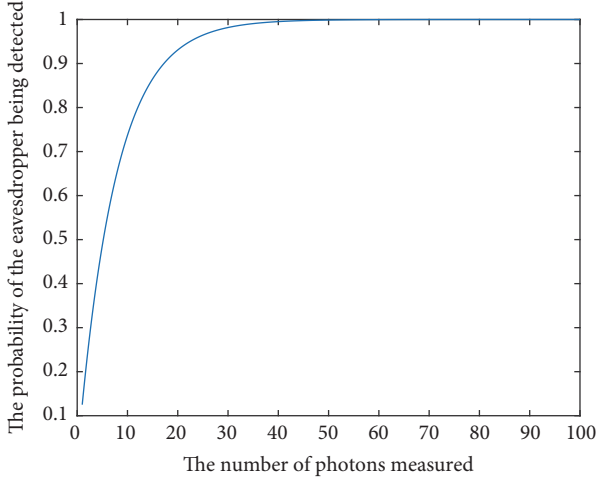


FIGURE 5: QKD protocol in noise free channel.

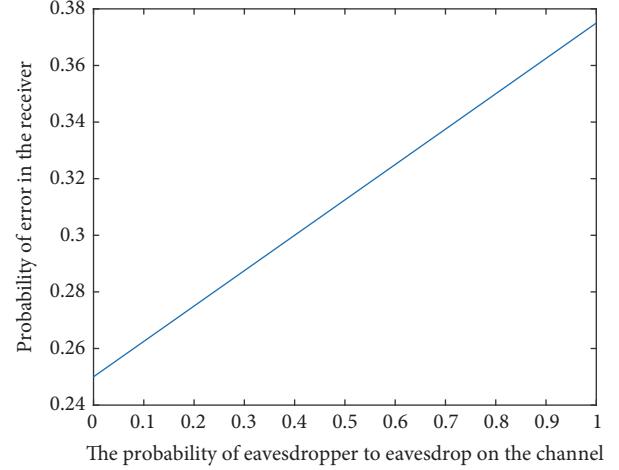


FIGURE 7: The effect of eavesdropping on the rate of error.

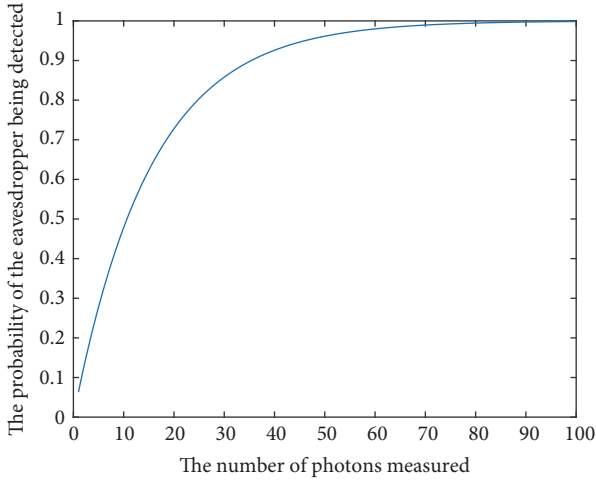


FIGURE 6: QKD protocol with 30% noise.

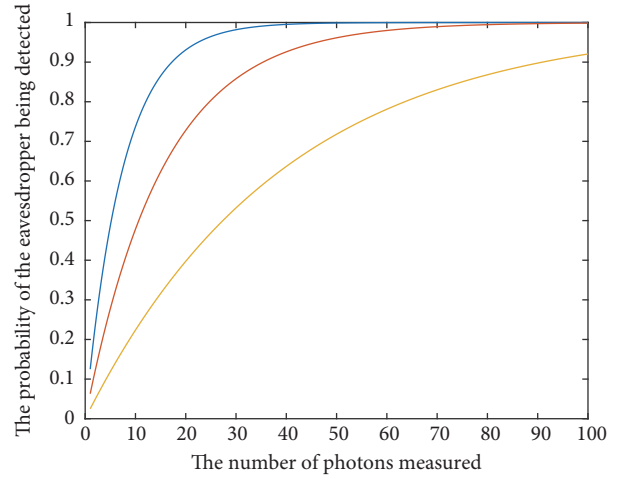


FIGURE 8: The eavesdropper detects the channel with different probability.

monitors the channel in the possibility of 50% and 20%, respectively. From these three curves, we can observe that regardless of probability of the eavesdrop monitoring the channel, the probability of him/her being detected is nearly 100% as the number of transmitted bits is rising.

From the above discussion, we can conclude that the quantum cryptography offers unconditional security and the sniffing detection properties for secure communication. These properties can ensure security for cyberspace in the future Internet.

5. Conclusion

Based on quantum mechanics and classical cryptography, quantum cryptography is a novel one in the field of cryptography. Compared with classical cryptography, its ultimate advantages are the unconditional security and the sniffing detection. These characteristics can solve cyberspace security critical problem for the future Internet. In particular, quantum cryptography provides security for various applications (e.g., Internet of things and smart cities [40]) in cyberspace

for the future Internet. Our experimental analysis results show the unconditional security and sniffing detection of quantum cryptography, which makes it suitable for future Internet.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Science Foundation of China under Grants no. 61672295 and no. 61672290, the State Key Laboratory of Information Security under Grant no. 2017-MS-10, the CICAET fund, and the PAPD fund.

References

- [1] L. Strate, "The varieties of cyberspace: Problems in definition and delimitation," *Western Journal of Communication*, vol. 63, no. 3, pp. 382–412, 1999.
- [2] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 2402–2415, 2017.
- [3] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, 2017.
- [4] T. Zhou, L. Chen, and J. Shen, "Movie Recommendation System Employing the User-Based CF in Cloud Computing," in *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 46–50, Guangzhou, China, July 2017.
- [5] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [6] 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.
- [7] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.
- [8] Y.-M. Tseng, "An efficient two-party identity-based key exchange protocol," *Informatica*, vol. 18, no. 1, pp. 125–136, 2007.
- [9] J. Shen, T. Miao, Q. Liu, S. Ji, C. Wang, and D. Liu, "S-SurF: An Enhanced Secure Bulk Data Dissemination in Wireless Sensor Networks," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, vol. 10656 of *Lecture Notes in Computer Science*, pp. 395–408, Springer International Publishing, Cham, 2017.
- [10] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (SFCS '94)*, pp. 124–134, IEEE, 1994.
- [11] 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*, pp. 1–1.
- [12] A. Peres, *Quantum Theory: Concepts And Methods*, Springer Science & Business Media, 2006.
- [13] S. Wiesner, "Conjugate coding," *ACM SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983.
- [14] C. H. Bennett and G. Brassard, "WITHDRAWN: Quantum cryptography: Public key distribution and coin tossing," *Theoretical Computer Science*, 2011.
- [15] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Physical Review Letters*, vol. 67, no. 6, pp. 661–663, 1991.
- [16] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," *Physical Review Letters*, vol. 68, no. 21, pp. 3121–3124, 1992.
- [17] B. Huttner, N. Imoto, N. Gisin, and T. Mor, "Quantum cryptography with coherent states," *Physical Review A: Atomic, Molecular and Optical Physics*, vol. 51, no. 3, pp. 1863–1869, 1995.
- [18] D. Bruß, "Optimal eavesdropping in quantum cryptography with six states," *Physical Review Letters*, vol. 81, no. 14, pp. 3018–3021, 1998.
- [19] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, pp. 1–10, 2017.
- [20] C. Crépeau, "Quantum oblivious transfer," *Journal of Modern Optics*, vol. 41, no. 12, pp. 2445–2454, 1994.
- [21] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska, "Practical quantum oblivious transfer," in *Annual International Cryptology Conference*, pp. 351–366, Springer.
- [22] D. Mayers and L. Salvail, "Quantum oblivious transfer is secure against all individual measurements," in *Proceedings of the Workshop on Physics and Computation. PhysComp '94*, pp. 69–77, Dallas, TX, USA.
- [23] D. Mayers, "On the security of the quantum oblivious transfer and key distribution protocols," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 963, pp. 124–135, 1995.
- [24] S. Winkler and J. Wullschlegel, "On the efficiency of classical and quantum oblivious transfer reductions," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6223, pp. 707–723, 2010.
- [25] A. Chailloux, I. Kerenidis, and J. Sikora, "Lower bounds for quantum oblivious transfer," *Quantum Information & Computation*, vol. 13, no. 1-2, pp. 0158–0177, 2013.
- [26] M. Curty and D. J. Santos, "Quantum authentication of classical messages," *Physical Review A: Atomic, Molecular and Optical Physics*, vol. 64, no. 6, 2001.
- [27] B.-S. Shi, J. Li, J.-M. Liu, X.-F. Fan, and G.-C. Guo, "Quantum key distribution and quantum authentication based on entangled state," *Physics Letters A*, vol. 281, no. 2-3, pp. 83–87, 2001.
- [28] D. Zhang and X. Li, "Quantum authentication using orthogonal product states," in *Proceedings of the 3rd International Conference on Natural Computation, ICNC 2007*, pp. 608–612, China, August 2007.
- [29] G. Brassard and C. Crépeau, "Quantum bit commitment and coin tossing protocols in," in *Proceedings of the Conference on the Theory and Application of Cryptography*, pp. 49–61, Springer.
- [30] N. K. Langford, R. B. Dalton, M. D. Harvey et al., "Measuring entangled qutrits and their use for quantum bit commitment," *Physical Review Letters*, vol. 93, no. 5, Article ID 053601, pp. 1–53601, 2004.

- [31] G. Zeng and C. H. Keitel, "Arbitrated quantum-signature scheme," *Physical Review A: Atomic, Molecular and Optical Physics*, vol. 65, no. 4, 2002.
- [32] X. Lü and D. Feng, "An Arbitrated Quantum Message Signature Scheme," in *Computational and Information Science*, vol. 3314 of *Lecture Notes in Computer Science*, pp. 1054–1060, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [33] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, 2017.
- [34] W. Heisenberg, "Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik," in *Original Scientific Papers Wissenschaftliche Originalarbeiten*, pp. 478–504, Springer, 1985.
- [35] A. Peres and L. E. Ballentine, "Quantum Theory: Concepts and Methods," *American Journal of Physics*, vol. 63, no. 3, pp. 285–286, 1995.
- [36] A. K. Pati and S. L. Braunstein, "Impossibility of deleting an unknown quantum state," *Nature*, vol. 404, no. 6774, pp. 164–165, 2000.
- [37] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," *Physical Review Letters*, vol. 70, no. 13, pp. 1895–1899, 1993.
- [38] B. M. Terhal, D. P. DiVincenzo, and D. W. Leung, "Hiding bits in bell states," *Physical Review Letters*, vol. 86, no. 25, pp. 5807–5810, 2001.
- [39] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.
- [40] J. Shen, D. Liu, J. Shen, Q. Liu, and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive and Mobile Computing*, vol. 41, pp. 219–230, 2017.