Wireless Communications and Mobile Computing

Secure Computation on 4G/5G Enabled Internet-of-Things

Lead Guest Editor: Karl Andersson
Guest Editors: Ilsun You, Rahim Rahmani, and Vishal Sharma

Secure Computation on 4G/5G Enabled Internet-of-Things

Wireless Communications and Mobile Computing

Secure Computation on 4G/5G Enabled Internet-of-Things

Lead Guest Editor: Karl Andersson
Guest Editors: Ilsun You, Rahim Rahmani, and Vishal Sharma

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 3978193, 1 page
https://doi.org/10.1155/2019/3978193

Editorial

Secure Computation on 4G/5G Enabled Internet-of-Things

Karl Andersson,[1] **Ilsun You**,[2] **Rahim Rahmani**,[3] **and Vishal Sharma**[2]

[1] *Luleå University of Technology, Luleå, Sweden*
[2] *Soonchunhyang University, Asan, Republic of Korea*
[3] *Stockholm University, Stockholm, Sweden*

Correspondence should be addressed to Karl Andersson; karl.andersson@ltu.se

Received 30 April 2019; Accepted 30 April 2019; Published 2 June 2019

The rapid development of Internet-of-Things (IoT) techniques in 4G/5G deployments is witnessing the generation of massive amounts of data which are collected, stored, processed, and presented in an easily interpretable form. Analysis of IoT data helps provide smart services such as smart homes, smart energy, smart health, and smart environments through 4G and 5G technologies. At the same time, the threat of the cyberattacks and issues with mobile internet security is becoming increasingly severe, which introduces new challenges for the security of IoT systems and applications and the privacy of individuals thereby. Protecting IoT data privacy while enabling data availability is an urgent but difficult task.

Data privacy in a distributed environment like IoT can be attained through secure multiparty computation. An emerging area of potential applications for secure computation is to address privacy concerns in data aggregation and analysis to match the explosive growth of the amount of IoT data. However, the inherent complexity of IoT systems really complicates the design and deployment of efficient, interoperable, and scalable secure computation mechanisms. As a result, there is an increasing demand for the development of new secure computation methods and tools which can fill in the gap between security and practical usage in IoT.

The scope of this special issue is in line with recent contributions from academia and industry on the recent activities that tackle the technical challenges making computing secure on 4G/5G enabled Internet-of-Things. For the current issue, we are pleased to introduce a collection of papers covering a range of topics such as securely verifiable remote erasure schemes, multiuser identification algorithms, privacy-preserving shared storage, situational aware threat assessment, authorized client-side deduplication in cloud storage, radio environment map construction, analysis of the vulnerabilities of connected car environments, combat pollution attacks in 5G multihop networks, automatically traceback RDP-based targeted ransomware attacks, multiresolution face recognition through virtual faces generation, anonymous communication via anonymous identity-based encryption, and Secure Storage and Retrieval of IoT Data.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

Acknowledgments

As always, we appreciate the high quality submissions from authors and the support of the community of reviewers.

Karl Andersson
Ilsun You
Rahim Rahmani
Vishal Sharma

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 6219868, 13 pages
https://doi.org/10.1155/2019/6219868

Research Article

Toward Privacy-Preserving Shared Storage in Untrusted Blockchain P2P Networks

Sandi Rahmadika [1] **and Kyung-Hyune Rhee** [2]

[1] *Interdisciplinary Program of Information Security, Graduate School, Pukyong National University, Republic of Korea*
[2] *Department of IT Convergence and Application Engineering, Pukyong National University, Republic of Korea*

Correspondence should be addressed to Kyung-Hyune Rhee; khrhee@pknu.ac.kr

Received 18 January 2019; Revised 9 March 2019; Accepted 2 April 2019; Published 16 May 2019

Academic Editor: Ilsun You

The shared storage is essential in the decentralized system. A straightforward storage model with guaranteed privacy protection on the peer-to-peer network is a challenge in the blockchain technology. The decentralized storage system should provide the privacy for the parties since it contains numerous data that are sensitive and dangerous if misused by maliciously. In this paper, we present a model for shared storage on a blockchain network which allows the authorized parties to access the data on storage without having to reveal their identity. Ring signatures combined with several protocols are implemented to disguise the signer identity thereby the observer is unlikely to determine the identity of the parties. We apply our proposed scheme in the healthcare domain, namely, decentralized personal health information (PHI). In addition, we present a dilemma to improve performance in a decentralized system.

1. Introduction

Since being introduced to the public through the rise of Bitcoin, blockchain has attracted a lot of attention among researchers, especially the way it deals in a transaction without involving the third parties. The blockchain technology reduces the transaction costs and it improves the efficiency and reliability of the decentralized system in general [1]. Due to its merits, blockchain has been developed in various fields of study such as logistics, e-commerce, trading activity, and healthcare, to name a few. Blockchain in the healthcare area is growing rapidly [2] as a future trend of substantial impact [3]. It aims at improving the quality of service and maintaining the integrity of information. Blockchain must be mature enough in all aspects, especially in security matters before blockchain being applied to a sensitive system (healthcare domain) [4] since it consists of valuable data for patients, providers, and all parties involved.

The shared storage between healthcare providers and the patients is one of the factors that must be considered when determining the scheme of the decentralized healthcare system. The surveys indicate that users often do not fully trust to store their data to third parties [5]. There are decentralized storage providers that provide the alternative services to protect the user's privacy such as Freenet [6] and GNUnet [7]. However, those services still have some drawbacks such as free-rider problem. More precisely, the provider is less motivated to keep improving system reliability due to the fact that there is no significant benefit for the provider to preserve the users' data. Apart from the free-rider problem, the main issue in the decentralized shared storage is related to the privacy of users [8]. An observer may be able to see the contents of online activities or metadata of the user since the data is publicly available.

To deal with the issues, we propose a model of the decentralized shared storage system on the blockchain that provides the privacy of the users without the involvement of third parties. Ring signature algorithm is applied to disguise the original identity of the signer. The parties involved use signatures on behalf of a group; hence, the original identity of the signer is unknown called signer ambiguous. In order

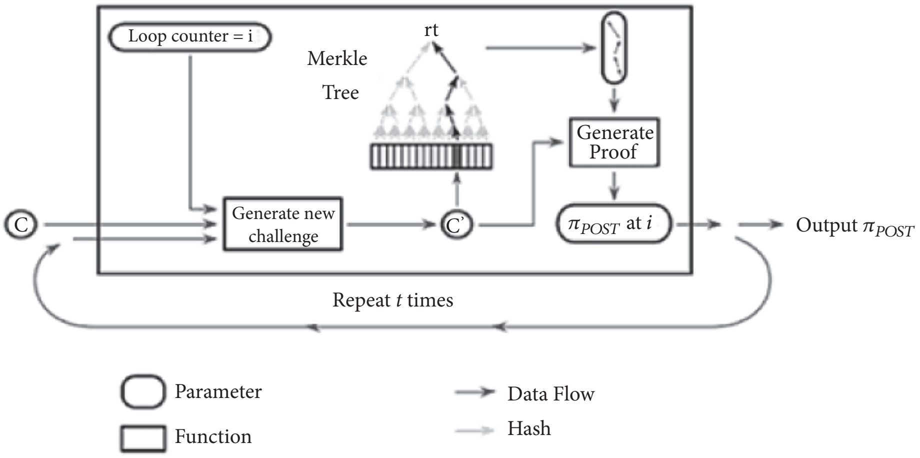


Figure 1: The iterative proof of Filecoin, adapted from [9].

to keep the identity of the parties to remain untraceable, one-time use address (from the stealth address) is adapted so that the observer cannot link the user address based on the transaction that has been carried out in advance.

The predecessor approaches to design the sharing storage system in the blockchain have been started by researchers lately such as Storj [10], storage with financial incentives, and Filecoin (see Figure 1) which generates a proof-of-spacetime (PoST) for the replica [11]. This paper presents the key concepts in the decentralized sharing storage in the healthcare system. The personal health information of the patient is propagated in the peer-to-peer blockchain network and the data are stored in a storage provider. The model of decentralized healthcare system comes from our previous research [12]. The privacy-preserving for the user is beyond the topic at the time. This paper is ongoing research and interrelated with our previous research.

The structure of the paper is organized as follows. Section 2 describes the background and core system component such as the ring signature, CryptoNote, and one-time use address (stealth address). Section 3 presents the system model of decentralized PHI data as well as the concept of ring confidential transaction. Section 4 presents the system analysis including the dilemma of reparameterizing propagation time and block size in order to improve the performance in a transaction. The limitations and future work are written in Section 5. Finally, Section 6 concludes the paper.

2. Background

In this section, we briefly present the essential information of ring signature algorithm, CryptoNote protocol, one-time use transaction address, and stealth address which are basic components for the privacy-preserving model in our system.

2.1. The Essential of Ring Signature. Ring signature is first introduced by Rivest et al. [13] in 2001 through the paper entitled "How to Leak a Secret". The idea of a ring signature originates from the concept signature group proposed by Chaum et al. [14] which in the group signature each member agrees to sign the message. In short, the data is signed on behalf of the group. In the group signature, there is a manager who organizes each activity in a group. As opposed to the group signature, the ring algorithm does not possess a manager in the process and neither have special requirements for creating groups as shown in Figure 2.

In order to form a signature group, the signer requires public keys P_k knowledge from prospective members. The selected public keys are encrypted by using a trapdoor permutation function (RSA, Rabin, and Diffie-Hellman). Due to the nature of the ring signature protocol, there are no specific rules for the number of members in a group. The standard procedure of the ring signature protocol can be defined as follows:

(i) ***Sign*** $\boldsymbol{\sigma}(msg, P_{sn}, P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn})$. The signature consists of the public keys $(P_{sn}, P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn})$ of the members for every message msg concatenated with the secret key P_{sn} of the signer to produce a signature σ.

(ii) ***Verify***$(\boldsymbol{msg},\boldsymbol{\sigma})$. The verification process can be interpreted as accepting a group signature σ which consists of public keys of all the possible signers along with the message msg. The final output is $true$ or $false$.

Generating a ring signature can be used directly by the signer without involving the group manager. The initial step is the signer computes the symmetric key Sym_k as the hash value of the message msg to be signed as $Sym_k = h(msg)$. The more complicated variant generates $Symk$ as $h(msg, P_{k1}, P_{k2}, P_{k3}, \ldots, P_r)$. However, the simpler creation is also secure. An initial random value R_v (or "glue") is chosen by the signer uniformly at random form $\{0,1\}^b$, where 2^b is some power of two which is larger than all modulo $n_i's$. Furthermore, the signer selects the number of signatures x_i from the ring members $1 < i < r, i \neq s$, where r is the ring members and s is the order

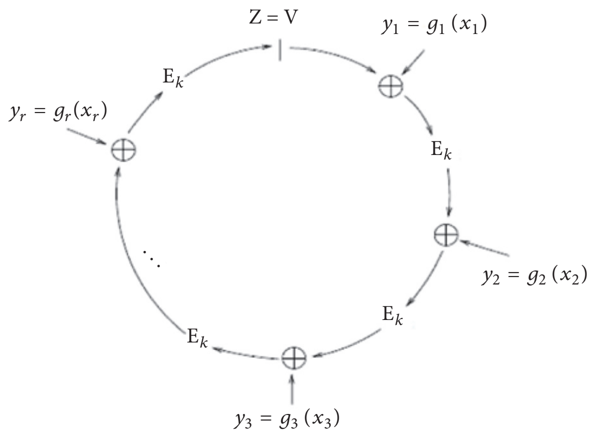


Figure 2: Ring signature algorithm which is defined by any member of a group of parties each having keys.

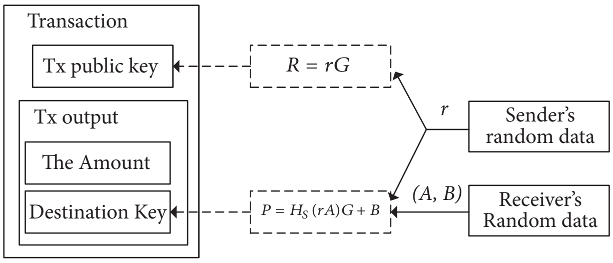


Figure 3: The structure of CryptoNote standard transaction.

of the member (s-th member) who is the actual signer. Hence, the signature gets a new value which is signified by $y_i = g(x_i)$. Finally, the signature of the message *msg* can be defined as $(P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn}; R_v; x_1, x_2, \ldots, x_r)$. The verification process is straightforward by describing the message received from the sender via secure channel $(P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn}; R_v; x_1, x_2, \ldots, x_r)$.

2.2. CryptoNote Protocol. The use of the ring signature algorithm in blockchain transaction was first introduced in 2012 which is part of the CryptoNote protocol [15] and updated in 2013. The CryptoNote constructs the ring signature using the public key of the random addresses and it provides privacy for the patient by leveraging the stealth address protocol [16]. By doing so, the observer believes that the signer has a secret key that corresponds to the cryptocurrencies, but the observer cannot determine the specific identity of the sender. However, the original ring signature protocol would allow double-spending attack in the blockchain transaction. The original ring signature protocol cannot determine the origin of the coin due to the fact that there is no marker if the transaction has been sent to the recipient. As a solution, CryptoNote provides one-time use ring signatures and key image as a marker.

The key image acts as a unique marker for every transaction. The key image gives the information about the transaction with a particular signature σ_n. If the same signature is used more than once, the miners will reject the transaction. In other words, any attempt to double-spend will indicate the use of the same key image. The destination of each CryptoNote output is a public key $(P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn})$ which is derived from the recipient's address combined with the sender's random value. In this regard, the sender asks the recipient's public key (A, B) via secure channel and the sender generates the one-time public key $P = H_s(rA)G + B$ as can be seen in Figure 3. Based on the key image that recipient belongs to, the recipient checks every passing transaction using his/her secret key (a, b) and calculates $P' = H_s(aR)G + B$, where H_s is a cryptographic hash function $\{0, 1\}*$ and G is a base point. Finally, the recipient can define $aR = arG = rA$ and $P' = P$. In addition, the recipient can recover the corresponding one-time secret key $x = H_s(aR) + b; P = xG$. By signing x transaction, the recipient can spend this output at any time. The security from this protocol is an untraceable transaction for the observer since the incoming message received by a recipient associated with one-time public keys (unlinkable).

2.3. One-Time Use Transaction and Stealth Address. We first present the one-time use transaction in general and we briefly describe the drawbacks of one-time use address in the blockchain transaction. To address the problem, we use the stealth address to protect the recipient information in our system model.

2.3.1. One-Time Use Transaction. Using the same address for each transaction on the blockchain network allows observers to track transactions to the original sender even though the

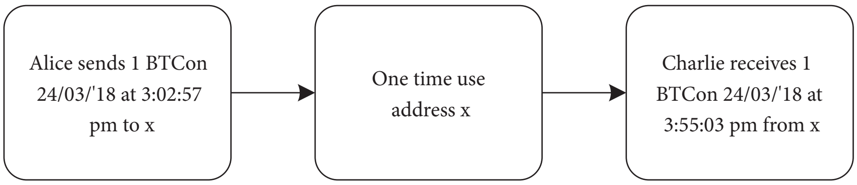

Alice sends 1 BTCon 24/03/'18 at 3:02:57 pm to x
One time use address x
Charlie receives 1 BTCon 24/03/'18 at 3:55:03 pm from x

Figure 4: One-time use transaction.

address is in the pseudonymous form. Since the new address does not have a track record in the blockchain metadata, the observer is unlikely to track information from a transaction. However, one-time use transaction address provides privacy for the users by extending the address to the counter-parties which detail information of the transaction still publicly available on the blockchain network. Roughly speaking, the sender enables the recipient to spend the funds that he just received even though the recipient identity remains hidden as can be seen in Figure 4.

For instance, an address X is a one-time address, but the observer has knowledge that Alice sent 1BTC to X and Charlie received from X. It can be used by the observer to infer that X corresponds to Alice and Charlie. By gathering the details on where Charlie's fund originated and where Alice's funds were passed on to, it could avail deanonymize the one-time use transaction. This scheme is also called *transaction graph analysis*. Therefore, in order to develop privacy on the decentralized blockchain system, a new protocol is necessary such as stealth address protocol. In other words, it is a security protocol that has become ubiquitous almost by stealth [17].

2.3.2. Stealth Address. One-time use address for transaction might be inconvenient to manage since the new address must be generated by the recipient for every transaction that will be carried out. Unlike the one-time use payment address, the stealth addresses get rid of this requirement. In short, stealth addresses can be generated as follows:

(i) The recipient generates a parent key pair $Pub(A, B)$ and publishes his/her public key (the published key is called *stealth address*).

(ii) The sender will be able to use the stealth address of the recipient to commit a new one-time use payment address for a particular transaction.

(iii) At this stage, the recipient uses their parent private key $P_s(a, b)$ which is generated in advance to spend the funds received. The generating process of stealth address can be seen in Figure 5.

The addresses are generated using trapdoor permutation such as Diffie-Hellman key exchange protocol [18]. By leveraging the stealth address protocol, the system allows eliminating the possibility for observers to link a one-time address to another. In this sense, the observer cannot determine the recipient's address and is unlikely to link the transaction to the recipient due to the new address being generated based on his/her stealth address for every transaction. Stealth address protocol is used in CryptoNote and Zcash combined with various other techniques such as zk-SNARKs in order to provide the stronger privacy for the user in the decentralized blockchain network. The cryptographic approaches can prevent ad hoc networks against external attackers using node authentication and data encryption [19].

3. Our System Model

In this section, we elaborate the model of decentralized PHI data (the model is from our prior work), the sequences of the standard blockchain transaction, the group of ring signature, and the ring confidential transaction of PHI.

3.1. Decentralized PHI Data. A decentralized personal health information model originally came from our previous research [12]. In our predecessor work, blockchain technology is used to manage the personal health information (PHI) data of the patient which obtain from several healthcare providers as can be seen in Figure 6. Based on the decentralized PHI model, we conducted testing in order to find out information about data communication in peer-to-peer networks (see Figure 7). The results obtained show the high level of success in sending data among the parties in the blockchain network that reaches 100% and the average of propagation message is 1:18ms for 100 bytes of Internet Control Message Protocol Echo. By leveraging the model of our previous research, the patients and the healthcare providers allow to collect effectively the PHI data onto a single view with integrity guarantee. Data integrity is essential for the patient in the blockchain network since it is a fundamental component of information security which verifies that the data has remained unaltered in transit from creation and reception [20]. By design, blockchain is tamper-proof, immutability (the data stored are unchanging over time or unable to be changed) so it is suitable for managing sensitive data such as personal health information, digital medical record, and other similar data.

In the decentralized healthcare system model, the patient and the providers are on the same blockchain network. The healthcare providers preserve a diagnosis from the patient and then store it into sharing storage right after the data is confirmed by the miner. The patient afterwards enables to find the data stored by the provider in the storage by searching one by one based on the patient's public key $Pub(A, B)$ attached into the transaction. The patient whose public key is attached to the transaction is the only party who knows the data stored in the storage because he/she has knowledge of the secret key $P_s(a, b)$ to access the PHI data.

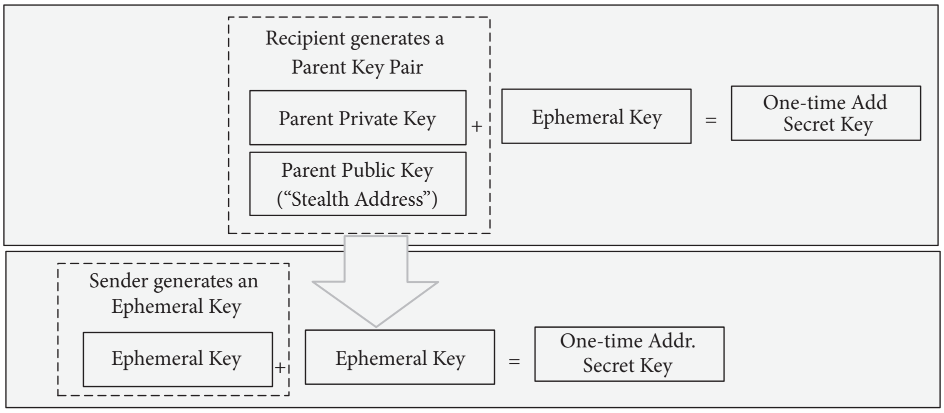

Recipient generates a Parent Key Pair
Parent Private Key
+
Ephemeral Key
=
One-time Add Secret Key
Parent Public Key ("Stealth Address")
Sender generates an Ephemeral Key
Ephemeral Key
+
Ephemeral Key
=
One-time Addr. Secret Key

Figure 5: Generating process of stealth address.

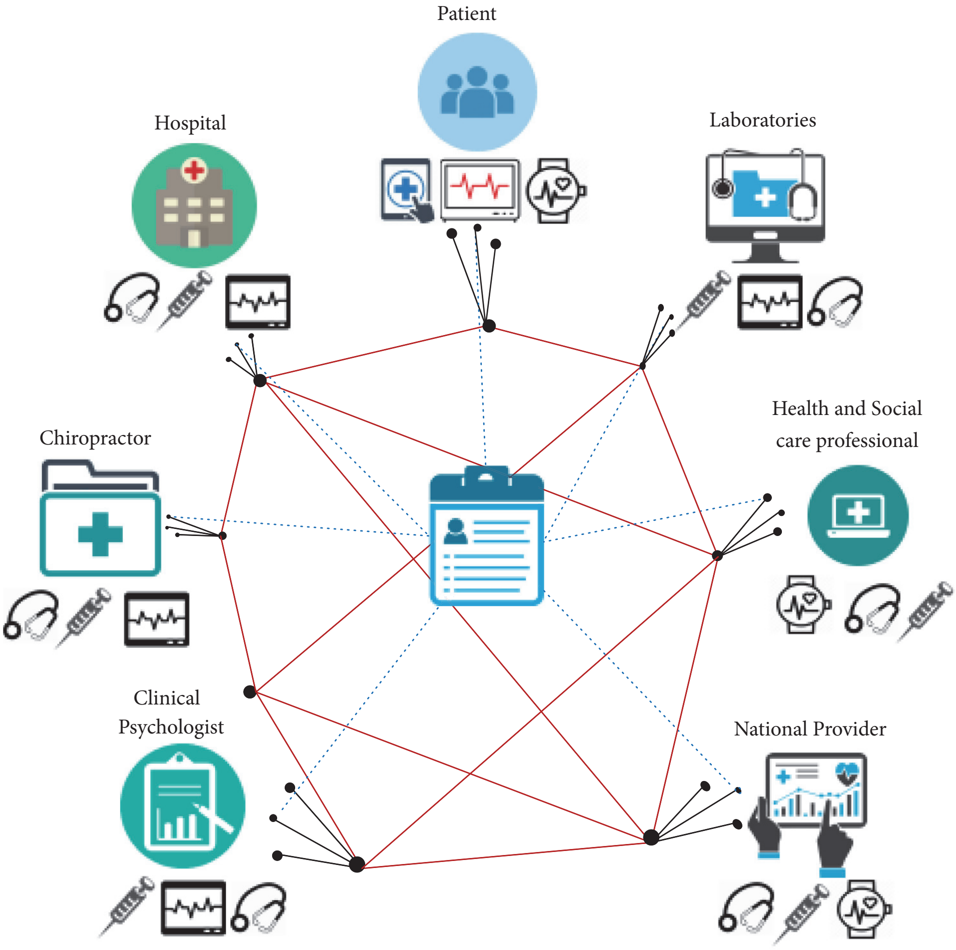


Figure 6: The model of decentralized personal health information data.

As with the security model in general, every party in the system has the unique parent keys. The public key is used to commit transactions, which later will be used to generate a stealth address for every transaction. A pair of keys is obtained from trapdoor permutation functions such as RSA algorithm and Rabin [21] which are generated beforehand. The parent keys of the parties can be defined as follows:

(i) The patient $(Pub_{\alpha}, Pr_{\alpha})$, the pair of patient's public key to commit the transaction can be defined as (A_{α}, B_{α}) and the pair of secret keys (a_{α}, b_{α}).

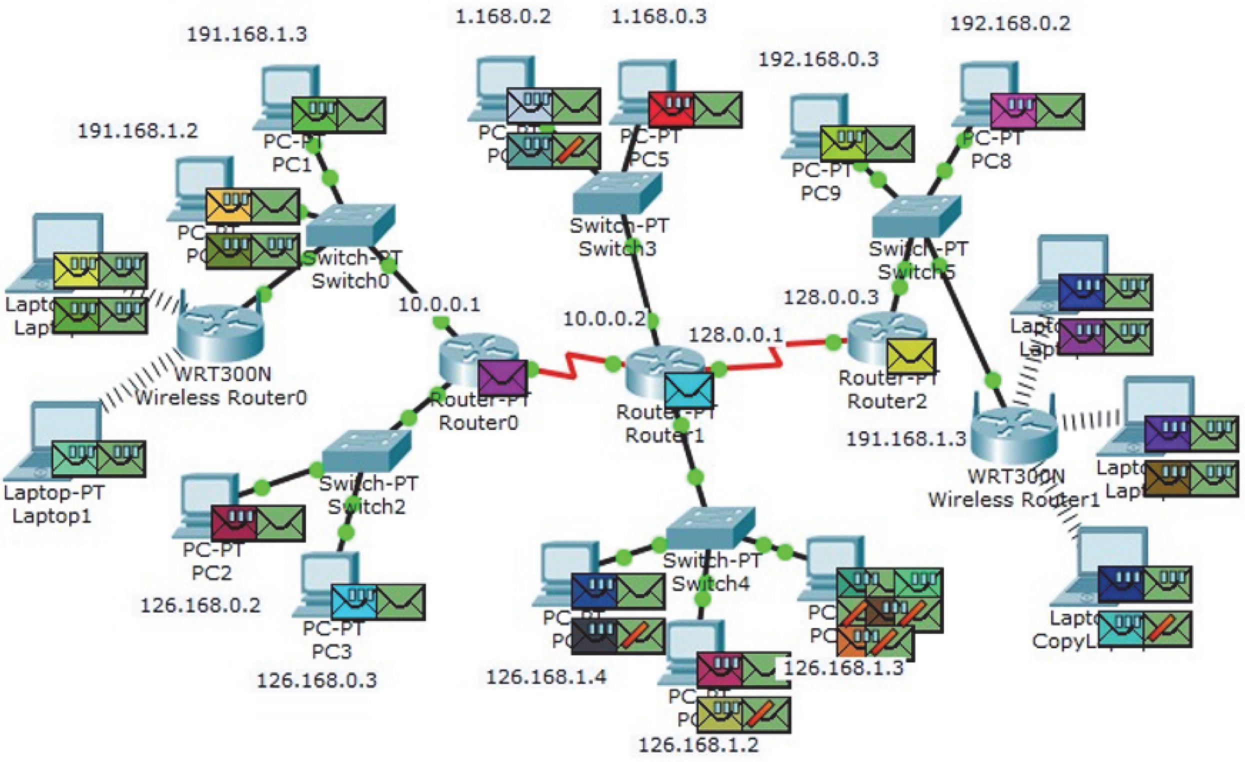


Figure 7: Propagating the personal health information data to the entire nodes.

(ii) Hospital (Pub_β, Pr_β), the pair of hospital's public key (A_β, B_β) and the pair of secret keys(a_β, b_β).

(iii) Chiropractor (Pub_γ, Pr_γ), Chiropractor's public key (A_γ, B_γ) and the pair of secret keys(a_γ, b_γ).

(iv) Clinical psychologist (Pub_δ, Pr_δ), clinical psychologist's public key (A_δ, B_δ) and the pair of secret keys (a_δ, b_δ).

(v) National provider $(Pub_\varepsilon, Pr_\varepsilon)$, the public key of national provider $(A_\varepsilon, B_\varepsilon)$ and the pair of secret keys $(a_\varepsilon, b_\varepsilon)$.

(vi) Health and social care provider (Pub_ζ, Pr_ζ), health and social provider's public key (A_ζ, B_ζ) and the pair of secret keys (a_ζ, b_ζ).

(vii) Laboratories(Pub_η, Pr_η), the pair of public keys of laboratories can be defined (A_η, B_η) and the pair of secret keys (a_η, b_η).

$$P = H_s(rA_\alpha)G + B_\alpha \tag{1}$$

$$P' = H_s(a_\alpha R)G + B_\alpha, then \tag{2}$$

$$\begin{aligned} a_\alpha R &= a_\alpha rG = rA_\alpha; \\ P' &= P \end{aligned} \tag{3}$$

The sequence of a standard transaction in decentralized PHI data starts from the provider who wants to store the patient's data in the storage where the patient has published his address to the provider beforehand. The provider unpacks the address and gets the patient's public key (A_α, B_α). The provider picks a random $r \in [1, l-1]$: where l is a prime order of the base point G. The provider then generates a one-time public key based on the pair public key (1) of the patient (the process is signified by Figure 3).

The patient and the healthcare providers possess a pair of public keys (A_n, B_n) for different purposes. The first public key A_n is used to generate a one-time public key, whilst the public key B_n is attached to the transaction as the tracking value used by the patient to find the data addressed to him/her. The provider uses P as a destination key for the output and attaches the new value $R = rG$ into the transaction. The PHI data with attachments to P and R values are stored into shared storage after being validated by the miner. The patient later checks every transaction using his private key (a_α, b_α) and calculates the new value P' (2). Finally, the patient compares the value P received with the value P' decrypted (3).

3.2. The Group of Ring Signature. In the decentralized PHI system, there is a group ring signature consisting of patient and several healthcare providers. In order to generate a group, the system does not demand special requirements and also unlikely require a manager to manage the group. All that is needed to create a ring signature group is the public key of each party $(Pub_\alpha, Pub_\beta, Pub_\gamma, Pub_\delta, Pub_\varepsilon, Pub_\zeta$, and $Pub_\eta)$.

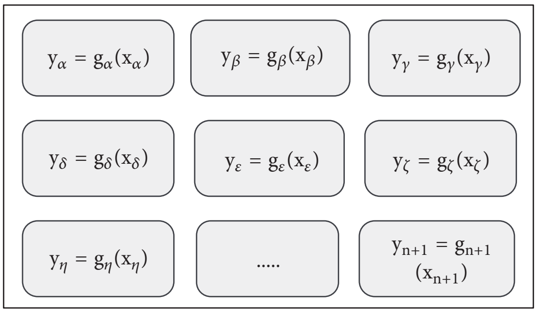

$y_\alpha = g_\alpha(x_\alpha)$
$y_\beta = g_\beta(x_\beta)$
$y_\gamma = g_\gamma(x_\gamma)$
$y_\delta = g_\delta(x_\delta)$
$y_\varepsilon = g_\varepsilon(x_\varepsilon)$
$y_\zeta = g_\zeta(x_\zeta)$
$y_\eta = g_\eta(x_\eta)$
.....
$y_{n+1} = g_{n+1}$ (x_{n+1})

FIGURE 8: The group of ring signature which is derived from the public keys of the member.

Once a ring signature group has been generated, all members of the group are allowed to use the signature and combine it with the private key of the sender. It grants users fine-grained control over the level of anonymity associated with a certain signature [22].

The signer enables to choose the number of signatures that they want to use in the transaction to provide an ambiguous signer. Later, the public key is used for encryption $y_n = g_n(x_n)$ as shown in Figure 8. Intuitively, g_n is defined by the extended trapdoor permutation function such as RSA and Rabin algorithm. In practice, for Rabin's functions $g_n(x_n)$ extends to $f_i(x_i) = x_i^2 \bmod n_i$ over $\{0,1\}^b$: where 2^b is the power of two which is larger than all modulo $n_i{}'s$. The bucket consists of b-bit numbers $w = q_i n_i + r_i$: where $r_i \in \mathbf{Z}^*_{\boldsymbol{n_i}}$ and $(q_i + 1)n_i \leq 2^b$. For any b-bit numbers w is nonnegative integers q_i and r_i. The bucket values are mapped by the extended Rabin mapping g_n. So, the value of $g_i(w)$ is signified by (4).

$$g_i(w) = \begin{cases} q_i n_i + f_i r_i & \text{if } (q_i + 1) n_i \leq 2^b \\ w & \text{else} \end{cases} \tag{4}$$

Intuitively, $g_i(w)$ is a permutation function over $\{0,1\}^b$ which is also a one-way trapdoor function because only a person knows the inverted value of f_i for a given input. Therefore, we can define the public keys for the prospective members as follows:

(i) The patient ⟵ $y_\alpha = g_\alpha(x_\alpha)$.

(ii) Hospital ⟵ $y_\beta = g_\beta(x_\beta)$: Chiropractor ⟵ $y_\gamma = g_\gamma(x_\gamma)$; whilst, the clinical psychologist ⟵ $y_\delta = g_\delta(x_\delta)$.

(iii) National provider ⟵ $y_\varepsilon = g_\varepsilon(x_\varepsilon)$: health and social care provider ⟵ $y_\zeta = g_\zeta(x_\zeta)$; Laboratories ⟵ $y_\eta = g_\eta(x_\eta)$.

(iv) For additional members of a group, it can be generated at any time as long as the public key of the new member is known $y_{\{n+1\}} = g_{\{n+1\}}(x_{\{n+1\}})$;

$$R_{sg} = y_\alpha \oplus y_\beta \oplus y_\gamma \oplus y_\delta \oplus y_\varepsilon \oplus y_\zeta \oplus y_\eta \oplus \ldots \oplus y_{\{n+1\}} \tag{5}$$

As can be seen in (5), a group of ring signature is constructed based on encryption of the member's public key. By design, the sender signs the message for the individual transaction using the signature of the group without a single group manager involved. Because the sender uses the signature of the group, the observer cannot judge the identity of the real sender for the corresponding transaction. The sender enables to choose the number of signatures that he/she wants to use in the transaction. The signature of a group can be used at any time in a transaction without having permission from the owner of the key. For instance, in a particular transaction the provider y_β uses the following keys to sign a message $h(msg, y_\gamma, y_\delta, y_\alpha$ and his key $y_\beta)$.

3.3. Ring Confidential Transaction of PHI.

3.3. Ring Confidential Transaction of PHI. Ring confidential transaction (RingCT) is used in Monero cryptocurrency [16] in order to improve the privacy of the users. Intuitively, the ring confidential transaction aims to hide the value of the actual amount in a transaction by combining the value of the current transaction with the value of the predecessor transactions. By doing so, the observer cannot tell the exact value of a transaction. The value of the transaction can be the number of coins sent or other data depending on the type of system that applies RingCT [23]. Unlike in the Monero transaction, the value transaction in the Bitcoin is publicly available in the plaintext. The observer might be able to analyse the transaction values for certain purposes in the particular period of time. Therefore, public data will be dangerous if misused maliciously.

$$RCT = txid\,[(8) \parallel (2) \parallel (5) \parallel (11) \parallel (12) \parallel (15)] \tag{6}$$

Suppose all outputs in Figure 9 exist, whilst the transaction that provider enables to spend is highlighted in green. Whenever the provider creates a transaction, he uses a RingCT to disguise which input is actually being spent. In this regard, the provider combines the current transaction with the previous transactions (highlighted in gray). The confidential ring signature for the transaction is shown in (6). The provider allows using the RingCT directly without permission and node manager involvement. As well as the group signature, the sender is free to use the value of previous transactions in order to disguise the value of the current transaction. By leveraging the model, it is possible to create privacy-preserving for users in the decentralized peer-to-peer shared storage in healthcare area with the following main objectives:

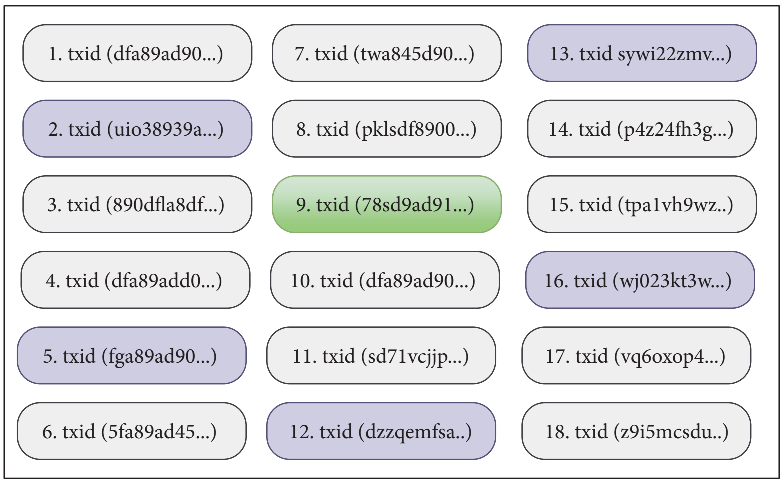


Figure 9: Ring confidential transaction based on the prior transactions.

(i) **Untraceability** with the goal to protect the sender's information in the form of the address *(public key)*. The observer cannot trace where the coin was received and where the coin originated from.

(ii) **Unlinkability** to preserve the recipient's identity. This can be achieved by using a stealth address where the recipient makes two public keys that are used to create a one-time address and the other as the view key.

(iii) **Confidential Values** to disguise the value of a transaction. The value of the current transaction is combined with the values of the transactions that have been carried out previously so it becomes an obscured transaction.

4. Systems Analysis and the Dilemma

The main protocols for building privacy-preserving for blockchain shared storage model have been discussed in previous section. In this section, we present the relation between each protocol. The combination of the protocols provides a system that enables protecting the privacy of users in the decentralized shared storage. The procedures are displayed gingerly, starting from generating the members of ring signature through to mechanism of storing data in the blockchain shared storage. At the end of this section, we elaborate on some dilemmas.

We demonstrate a case study in which one of the providers (hospital) wants to store diagnostic data of the patient into the decentralized shared storage. The provider has known the address *(public key)* of the patient beforehand. In this sense, the hospital acts as the sender whilst the patient acts as the recipient of the PHI message. The provider attaches the address of the patient in the form of a *view key* for each transaction so that only patients likely enable to track data stored in shared storage using his/her knowledge. The miners have a duty to validate the data from the sender before being saved into shared storage, yet they are responsible for adding blocks to the blockchain network. The miner gets a reward after successfully adding the new block as with the blockchain system in general. Algorithm 1 is a description of the whole system process starting with making public keys through to data stored in shared storage. The process sequence for privacy-preserving shared storage in untrusted blockchain P2P networks as follows:

(i) The party possesses a pair of parent keys (Pub_{key}, Pr_{key}) that have been generated beforehand based on trapdoor permutation function such as RSA, Rabin, or Diffie-Hellman algorithm. The patient is the recipient(Pub_{α}, Pr_{α}), whilst the hospital is the sender (Pub_{β}, Pr_{β}) in this case study.

(ii) The hospital is the sender of the patient's diagnosis data (in this case). The hospital constructs a ring signature group based on public key from providers and patient that has been known previously. The hospital also added its public key to the group.

$$\begin{aligned} R_{sg\beta} = {} & g_{\alpha}\left(x_{\alpha}\right) \oplus g_{\boldsymbol{\alpha}}\left(x_{\beta}\right) \oplus g_{\boldsymbol{\gamma}}\left(x_{\gamma}\right) \oplus g_{\delta}\left(x_{\delta}\right) \\ & \oplus g_{\varepsilon}\left(x_{\varepsilon}\right) \oplus g_{\zeta}\left(x_{\zeta}\right) \oplus g_{\eta}\left(x_{\eta}\right) \oplus \ldots \\ & \oplus g_{n+1}\left(x_{n+1}\right) \end{aligned} \tag{7}$$

(iii) When the group of ring signature $R_{sg\beta}$ is generated, the sender enables to use the signature of each group member without having to get permission from the owner of public keys. In this case study, the hospital chooses to use all signatures from group members as shown in (7). The hospital can add new members at any time as long as the public key is known.

(iv) The patient as the recipient later makes the stealth address based on a pair of parent keys. The patient creates a pair of public keys $Pub_{\alpha}(A_{\alpha}, B_{\alpha})$ and sends it to the hospital via secure channel. The hospital generates a new one-time address based on stealth

```
Procedure $Shared\_Storage$:
Trapdoor Function: (parent keys)    $*$trapdoor permutation function e.g. RSA, Rabin
        $Patient \longleftarrow hash(Pub_\alpha, Pr_\alpha)$
        $Hospital \longleftarrow hash(Pub_\beta, Pr_\beta)$    $*\forall$party has parent keys
        $Chiropractor \longleftarrow hash(Pub_\gamma, Pr_\gamma)$
        $Clc.Psychologist \longleftarrow hash(Pub_\delta, Pr_\delta)$
        $NationalProvider \longleftarrow hash(Pub_\varepsilon, Pr_\varepsilon)$
        $Social\ care \longleftarrow hash(Pub_\zeta, Pr_\zeta)$
        $Laboratories \longleftarrow hash(Pub_\eta, Pr_\eta)$
The sender creates a group of ring signature:
Procedure $Use\ Public\ Key\ \forall\ parties \in parentkeys$    $*$use public key of the parties as an input
Create RS:
        $R_{sgn} \longleftarrow g_\alpha(x_\alpha) \oplus g_\beta(x_\beta) \oplus g_\gamma(x_\gamma) \oplus \ldots \oplus g_{n+1}(x_{n+1})$
AddNewMembers:    $*$in case: add new members
        $\textbf{Update}\ R_{sgn} \longleftarrow Get\_NewPubkey \oplus PubKey_{(n+1)}$
end procedure
Procedure $Create\ StealthAddr(byRecipient)$:    $*$the recipient creates stealth address
        $ParentKey: PubKey_n \longrightarrow PubKey_n(A_n, B_n)$
        $Send(A_n) \longrightarrow to\_sender\_(via\_SecureChannel)$
        $(A_n, B_n) received \longrightarrow Create\_New\_OTP$    $*$sender creates new OTP for recipient
end procedure
Procedure $Confidential\ Ring\ Signature$:    $*$confidential ring signature as an option
        $if\ \forall txs\ BC\ use\ CRS\ \textbf{then return}$ True
        $NewValue \longleftarrow CurrentValue \oplus Prev.Value$
        $Include\ to\ the\ new\ tx \longleftarrow \textbf{NewValue}$
        $\textbf{elsereturn}$ False
end procedure
Procedure $Sign\ the\ PHI\ Data(Msg)$:    $*$in this case, the PHI data is from the hospital
        $\textbf{Get}\ Msg \in Diagnosis\_Data(Hospital)$
        $\textbf{Sign}\ \sigma \longleftarrow choose\_signer \in R_{sgn}$
        $\textbf{RingCT} \longleftarrow choose\_prev\_txs \in CRS$
        $\textbf{Msg} \longleftarrow attached' KeyImage'$    $*$KeyImage: to prevent storing the same data
        $\textbf{CurrMsg} \longleftarrow Sign\,\sigma \parallel RingCT \parallel KeyImage$
end procedure
Procedure $Send\ to\ Blockchain\ Network$:
        $\textbf{Get}\ CurrMsg \oplus OTP(from\ StealthAddr)$
        $\textbf{Send}\ to\ blockchain\ network$
        $\textbf{Wait}\ miner's\ confirmation$
        $\textbf{if}\ CurrMsg\ is\ valid\ \textbf{then return}$ True    $*$transaction is success
        $\textbf{Add}\ newBlock\ to\ BC\ network$
        $AddCurrMsg \longrightarrow Shared\_Storage$
        $\textbf{elsereturn}$ False
End
```

Algorithm 1: Shared Storage of PHI Data.

address received. Only patients can access data stored in shared storage by using his knowledge $Pr_\alpha(a_\alpha, b_\alpha)$.

$$OTP = H_s\left(rA_\alpha\right)G + B_\alpha \tag{8}$$

(v) The sender unpacks the address received which contains the public address of the recipient $Pub_\alpha(A_\alpha, B_\alpha)$. The sender picks the random value $r \in [1, l-1]$ to generate one-time public key and attaches B_α as a view address in the shared storage.

(vi) For certain types of data, the sender can use confidential ring signatures *(RingCT)* to disguise information of the data by adding value to transactions that have occurred before such as $RCT_\beta = txid(3) \parallel txid(7) \parallel txid(5) \parallel txid(9) \parallel txid(n)$.

(vii) The hospital signs the diagnosis data *msg* using the member key from the ring signature as follows: $\textbf{Sign}\,\boldsymbol{\sigma}(msg, y_\alpha, y_\beta, y_\gamma, y_\delta, y_\varepsilon, y_\zeta, y_\eta, \ldots, y_n)$ which consists of the public keys of the members $Pub_\alpha, Pub_\beta, Pub_\gamma, Pub_\delta, Pub_\varepsilon, Pub_\zeta$, and Pub_η.

(viii) Key image is attached by the sender to prevent double spending. It can be interpreted as a scheme to prevent the same data stored twice in blockchain storage. The hospital sends the following series of data to the blockchain network to be confirmed by miners: $Sign\,\sigma \parallel msg \parallel OTP \parallel RingCT \parallel keyimage$.

(ix) The diagnosis data from the hospital along with attachments of the files are then sent to the blockchain network to be verified by the miners before the data are stored in a decentralized shared storage. Whenever the data has been stored successfully, the patient traces one by one the transaction on the blockchain network until the patient finds his/her view keys B_n which correspond to the patient.

(x) The patient with his/her knowledge decrypts and compares the value of the data received with the decrypted value $a_\alpha R = a_\alpha rG = rA_\alpha; P' = P$.

The Algorithm 1 presents an overall model of the system which starts with generating key pairs for the parties, creating a ring signature group and stealth addresses, signing PHI data until data are confirmed and stored in the decentralized shared storage. The observer is unlikely to track the sender's transaction since the sender uses a signature on behalf of a group in which the sender's signature is obscured. Furthermore, the observer cannot associate one transaction with another, so it is indistinguishable whether the transaction was sent by the same sender. In other words, the identity of the sender remains a secret. Likewise, the identity of the recipient cannot be tracked by the observer nor malicious providers because the recipient sends the stealth address to the sender; thereafter, the sender creates a one-time address to protect the privacy of the recipient. The value of a PHI data is kept from being seen because it is combined with the value of previous transactions through the *RingCT*. The identity of the user along with the information of the transactions on the blockchain network is paramount; therefore, the decentralized shared storage systems need to manage the confidentiality as well as ensure the integrity of the PHI data.

Apart from the model that has been described, there is another important factor which plays the role to a decentralized system called transaction propagation. A numerous number of transactions are distributed to the entire nodes in the blockchain peer-to-peer network, resulting in propagation delay. The structure of a P2P network allows the peer to disseminate information to the other nodes that are connected to the sender [24]. There are several studies conducted by researchers to measure how effective the block distribution in the peer-to-peer networks such as experiments in which the goal is to find out the number of successful connections and experiments to determine the effect of the number of nodes against the block.

One among parameters that affect transaction propagation is block size. Block size can be understood as the maximum limit of a block to be filled up with various transactions on it. It also can be thought of as a bundle of transactions, with each block needing to get verification before it can be accepted by the network. Each block has its own size depending on the type of transaction called the block size. The maximum block size in Bitcoin stands at 1MB. Miners enable to choose the number of transactions to be processed in a block. If Bitcoin miners commit a transaction that exceeds the maximum limit, then the block will be rejected by the network. The motivation of block size is to prevent the attack such as denial-of-service attacks. The size of a block also affects the length of confirmation time. When a node receives a new transaction, the recipient confirms the validity of the block before accepting it. The duration of the confirmation process depends on the size of the block. By design, the larger the size of a block is, the longer it takes to confirm. Therefore, the size of a block plays an important role in the blockchain in general because it directly affects the delay time.

Propagation delay is inseparable from the size of a block. There is a correlation between block size and the propagation delay until the node receives the block. The larger the size of a block, the more transactions that can be done, yet it affects the propagation time and sacrifice the security. The influence of a block size to the propagation time can be seen in Figure 10 [25]. The block size in the transaction is varied up to 350 KB in order to find out how long it takes for the node to receive the block. To reach 25% of total transaction is visualized with a red line, 50% for the green line, and 75% for the blue line. The results are in line with the theory which states the larger the size of a block, the greater the propagation time.

We take measurements for orphaned blocks by following the withholding attack (selfish mining) strategy that was first discovered in 2014. In this study we do not elaborate on the details of withholding attack strategy, we recommend that readers refer to the references [26–28]. The intuition of this attack is to keep the finding block secret until the attacker's network becomes the longest chain in the blockchain network. For a particular condition, this attack does not possess any benefit because the block is stored in the attacker's network which is only known by the attacker (nonprofit). The attackers get the reward if only the block found is propagated to the public and get confirmed by other miners.

The simulation is carried out in order to know the performance of dishonest miners which follow the selfish mining protocol. The aims of this strategy are to discover the new block till becoming the longest chain and gaining the revenue after publishing the block to the public network [29]. In our setting, we arrange the selfish miners to compete with honest miners in 14 days to solve the proof-of-work and discover the new block. The maximum of mining power of selfish miner is 0.4 and it is running randomly from 0.0 through 0.4 within 14 days in the simulation as shown in Figure 11. There are 12 nodes of dishonest miners with different mining power from 0.0 to 0.4. We set the maximum number of mining power at 0.4 of total mining power in the network. Based on the simulation result, we conclude that whenever the dishonest miner has 0.322 mining power, it is enough to get the unfair revenue and allows gaining the revenue larger than it should be. In general, there are 51 new blocks successfully added as well as 4 orphaned blocks recorded. The average of block generation time is 7.72 minutes.

To the extent, we obtained the information related to the parameters that affect the propagation time in the blockchain peer-to-peer network based on our findings and the prior works of literature. There are numerous articles proposing the improvement of propagation delays by changing the network topology, minimizing the verification time of a transaction,

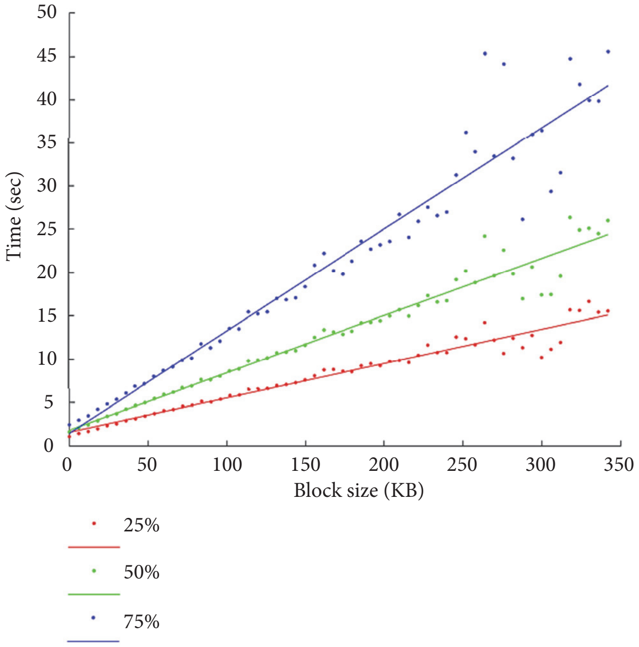


FIGURE 10: Relationship between block size and the time.

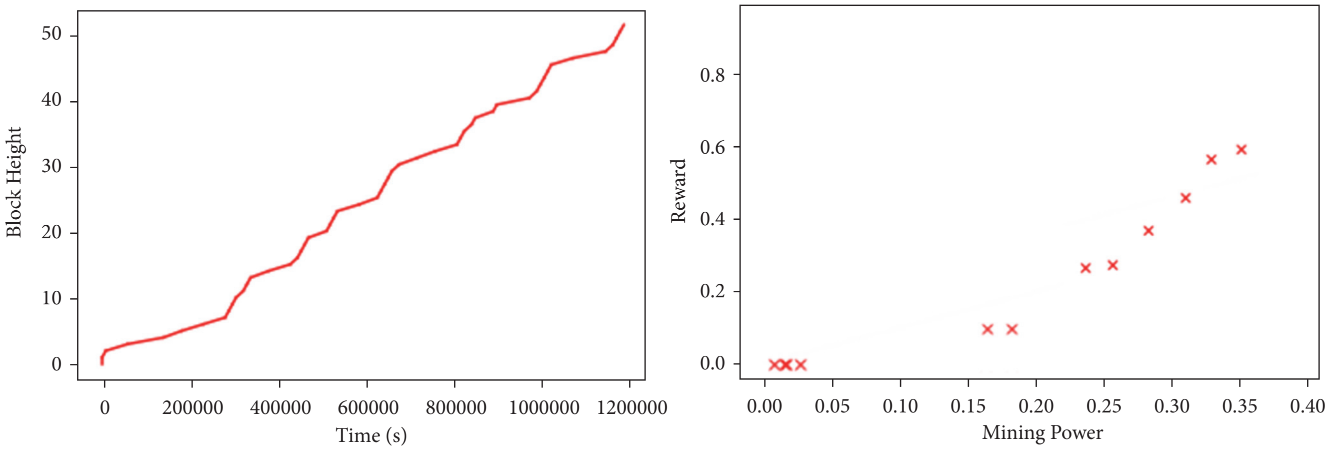


FIGURE 11: Block height against time in the P2P network (left); the performance of selfish mining attack (right).

and reconstructing the message exchange protocol in the blockchain network. Generally speaking, the presence of having the propagation of delays can cause a lot of damage in the blockchain network [30, 31].

There are many considerations to increase the effectiveness of the blockchain. Therefore, we select block propagation and block size parameters to be discussed as follows:

(i) *Speeding up the block generation*. In theory, it would be remarkably beneficial if the block generation time is resetting as fast as possible for each transaction so that each user will get faster payments. The propagation time includes the length of time for the distribution of transactions in the peer-to-peer network and the time for verification of a block. But the problem is that if the block generation time is speeding up, there will be a lot of orphaned blocks. It can be understood because each node will receive many new blocks that are spread through peer-to-peer networks. The *tie-breaking* protocol causes each node to only accept one valid block for the same type of transaction, so that it will reject transactions from other nodes that cause the orphaned block to appear. Due to many new orphaned blocks that will emerge, it will motivate rational miners to adopt the selfish mining strategy that rivals the main network and causes the forked chain [32].

(ii) *Decrease block generation*. Slowing down the block generation time for each transaction will reduce the speed of transactions on the peer-to-peer network. Positively, it gives some merits such as providing a better security system. Roughly this happens due to a decreasing number of orphaned blocks that will eliminate the selfish mining attack.

(iii) *Increasing the block size*. The bigger the capacity of a block, the more the transactions that can be filled up into the block. It will slow the propagation of every transaction to all nodes in the peer-to-peer network. The slow propagation time will cause new problems, namely, double-spending attack. Attacker could use the same coin for two or more different transactions. Slower propagation of blocks on the peer-to-peer network resulted in the fact that the block cannot be fully accepted by the nodes. As a result, when the transaction is received, the block will confirm that the block is valid even though the transaction has been used and confirmed by other nodes in the same network.

(iv) *Reducing the capacity of block size*. Because of only a few transactions that occur within a block, it will speed up the propagation time for every transaction. It causes many orphaned blocks to emerge and allow for the occurrence of selfish mining attacks that harm the system. Yet, this decision gives the advantages such as fast payments and fast transaction. However, it should ensure the immutability of the block and transaction [33].

5. Limited and Future Work

We assume that shared storage is interconnected to a blockchain network where miners have access into it. In terms of the type of shared storage, we do not define it in detail; instead we assume the shared storage has all the capacity needed to support the proposed system. One of the drawbacks of our system is related to the time needed by the recipient to find data in shared storage, because the recipient must seek for data one by one based on the key view, so that the patient observes each transaction in the blockchain. In the long run, this might become an obstacle if the blockchain network is expanding. There will be many transactions occurring so that it will be difficult for the recipient to monitor every transaction which belongs to him/her.

For future work, the model and capacity of shared storage need to be observed further. The access control in the shared storage is also essential to ensure that system keeps safe. Incentive mechanisms also need to be considered for the storage providers. It aims to motivate providers to contribute to protecting the privacy of the users as suggested by [34–36]. Furthermore, it is important to carefully consider the type of block to be used including parameters directly involved in the system such as block size and type of consensus, to name a few. Additionally, the consensus selection mechanism is paramount in the blockchain. For instance, a new method by expanding the Byzantine consensus via hardware-assisted secret sharing can solve the scalability problem in the blockchain [37].

6. Conclusions

The model of privacy in the blockchain peer-to-peer shared storage has been fully presented. The idea of ring signature combined with several protocols provides the solutions for privacy issues on the blockchain transaction. The identity of the sender and the recipient remains hidden, unlinkable, and untraceable from the observer. The key image is attached to prevent the same data from being stored multiple times, and it can be used as well to prevent double spending and data duplication. Based on our findings and information from several literature reviews, it can be said that increasing the performance of the decentralized blockchain requires a very deep analysis since it is directly related to the security. There are advantages and drawbacks for each decision taken. For future work, a scheme that provides incentives needs to be developed as a motivation to maintain the decentralized system.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-0-00403) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1D1A1B07048944).

References

[1] G. Karame and E. Androulaki, *Bitcoin and Blockchain Security*, Artech House Information Security and Privacy Series, 2016.

[2] R. J. Krawiec, D. Housman, M. White et al., "Blockchain: Opportunities for health care," in *Proceedings of the NIST Workshop Blockchain Healthcare*, pp. 1–16, 2016.

[3] D. E. O'Leary, "Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems," *Intelligent Systems in Accounting, Finance and Management*, vol. 24, no. 4, pp. 138–147, 2017.

[4] D. Yang, J. Gavigan, and Z. W. Hearn, "Survey of Confidentiality and Privacy Preserving Technologies for Blockchains," R3, pp. 1–32, 2016.

[5] H. Kopp, D. Mödinger, F. Hauck, F. Kargl, and C. Bösch, "Design of a privacy-preserving decentralized file storage with financial incentives," in *Proceedings of the 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017*, pp. 14–22, April 2017.

[6] T. W. Clarke, I. Sandberg, O. Wiley, and B. Hong, "A distributed anonymous information storage and retrieval system," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2001.

[7] E. Heilman, L. AlShenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "TumbleBit: an untrusted bitcoin-compatible anonymous payment hub," in *Proceedings of the 2017 Network and Distributed System Security Symposium*, 2017.

[8] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo, "CReam: a smart contract enabled collusion-resistant e-auction," *IEEE Transactions on Information Forensics*, 2018.

[9] J. Benet and N. Greco, "Filecoin: A Decentralized Storage Network," *Protoc. Labs*, 2018.

[10] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," Technical report, storj.io, 2014.

[11] B. Fisch, "Poreps: Proofs of space on useful data," Report 2018/678, Cryptology ePrint Archive, 2018.

[12] S. Rahmadika and K. Rhee, "Blockchain technology for providing an architecture model of decentralized personal health information," *International Journal of Engineering Business Management*, vol. 10, pp. 1–12, 2018.

[13] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2248 of *Lecture Notes in Comput. Sci.*, pp. 552–565, Springer.

[14] D. Chaum, E. van Heyst, and C. Science, "Group Signatures," in *Adv. Cryptology—EUROCRYPT'91*, vol. iii, pp. 257–265, 1991.

[15] N. Van Saberhagen, "CryptoNote v 2.0," *Self-published*, pp. 1–20, 2013.

[16] S. Noether, A. Mackenzie, and T. M. Research Lab, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–8, 2016.

[17] E. Gallery and C. J. Mitchell, "Trusted computing: Security and applications," *Cryptologia*, vol. 33, no. 3, pp. 217–245, 2009.

[18] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[19] A. A. Korba, M. Nafaa, and S. Ghanemi, "An efficient intrusion detection and prevention framework for ad hoc networks," *Information and Computer Security*, vol. 24, no. 4, pp. 298–325, 2016.

[20] S. Rahmadika, P. H. Rusmin, H. Hindersah, and K. H. Rhee, "Providing data integrity for container dwelling time in the seaport," in *Proceedings of the 6th International Annual Engineering Seminar, InAES 2016*, pp. 132–137, Indonesia, August 2016.

[21] K. Schmidt-Samoa, "A new rabin-type trapdoor permutation equivalent to factoring," *Electronic Notes in Theoretical Computer Science*, vol. 157, no. 3, pp. 79–94, 2006.

[22] A. Bender, J. Katz, and R. Morselli, "Ring signatures: stronger definitions, and constructions without random oracles," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 22, no. 1, pp. 114–138, 2009.

[23] K. Lee and A. Miller, "Authenticated data structures for privacy-preserving monero light clients," in *Proceedings of the 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018*, April 2018.

[24] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," in *Proceedings of the 1st International Conference on Peer-to-Peer Computing, P2P 2001*, pp. 101-102, August 2001.

[25] Y. Sompolinsky and A. Zohar, "Accelerating bitcoins transaction processing. fast money grows on trees, not chains," *Eprint.Iacr.Org*, 2014.

[26] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8437, pp. 436–454, 2014.

[27] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*, vol. 9603 of *Lecture Notes in Computer Science*, pp. 515–532, Springer, Berlin, Germany, 2017.

[28] E. Heilman, "One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract)," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8438, pp. 161-162, 2014.

[29] S. Rahmadika, B. J. Kweka, H. Kim, and K. Rhee, "A scoping review in defend against selfish mining attack in bitcoin," *IT Converg. Pract*, vol. 6, no. 3, pp. 18–26, 2018.

[30] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proceedings of the the 22nd ACM SIGSAC Conference*, pp. 692–705, Denver, Colo, USA, October 2015.

[31] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Čapkun, "On the security and performance of Proof of Work blockchains," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security, CCS 2016*, pp. 3–16, October 2016.

[32] D. T. T. Anh, M. Zhang, B. C. Ooi, and G. Chen, "Untangling blockchain: a data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[33] D. Mendez Mena, I. Papapanagiotou, and B. Yang, "Internet of things: survey on security," *Information Security Journal*, vol. 27, no. 3, pp. 162–182, 2018.

[34] O. Ersoy, Z. Ren, Z. Erkin, and R. L. Lagendijk, "Transaction propagation on permissionless blockchains: incentive and routing mechanisms," in *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 20–30, June 2018.

[35] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.

[36] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 6, pp. 17545–17556, 2018.

[37] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable Byzantine consensus via hardware-assisted secret sharing," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 68, no. 1, pp. 139–151, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 8913910, 11 pages
https://doi.org/10.1155/2019/8913910

Research Article

UAV-Undertaker: Securely Verifiable Remote Erasure Scheme with a Countdown-Concept for UAV via Randomized Data Synchronization

Sieun Kim,[1] **Taek-Young Youn**,[2] **Daeseon Choi**,[3] **and Ki-Woong Park**[4]

[1] *SysCore Lab., Sejong University, Seoul, Republic of Korea*
[2] *Electronics and Telecommunications Research Institute, Deajeon, Republic of Korea*
[3] *Dept. of Medical Information, Kongju National University, Kongju, Republic of Korea*
[4] *Dept. of Computer and Information Security, Sejong University, Seoul, Republic of Korea*

Correspondence should be addressed to Ki-Woong Park; woongbak@sejong.ac.kr

Received 31 January 2019; Revised 28 March 2019; Accepted 22 April 2019; Published 16 May 2019

Academic Editor: Ilsun You

Unmanned aerial vehicles (UAVs) play an increasingly core role in modern warfare, with powerful but tiny embedded computing systems actively applied in the military field. Confidential data, such as military secrets, may be stored inside military devices such as UAVs, and the capture or loss of such data could cause significant damage to national security. Therefore, the development of securely verifiable remote erasure techniques for military devices is considered a core technology. In this study, we devised a verifiable remote erasure scheme with a countdown-concept using randomized data synchronization to satisfy securely verifiable remote erasure technology. The scheme allows the GCS (Ground Control Station) to remotely erase data stored in the UAV, even on loss of communication, and returns proof of erasure to GCS after erasure. Our approach classifies the accumulated data stored in the UAV as a new data type and applies the characteristics of that data type to generate the proof of erasure. We select a small-volume data sample (rather than all of the data) and perform prior learning only on that sample; in this way, we can obtain the probative power of the evidence of erasure with a relatively small amount of traffic. When we want to erase data of 100 Mbytes of remote device, 100 Mbytes of data transfer is required for related work, whereas our system has data transfer according to the ratio of amount of randomly selected data. By doing this, communication stability can be acquired even in unstable communication situations where the maximum traffic can change or not be predicted. Furthermore, when the UAV sends the proof of erasure to the GCS, the UAV does its best to perform the erasure operation given its situation.

1. Introduction

In 2011, the US military lost control of an American Lockheed Martin RQ-170 Sentinel unmanned aerial vehicle (UAV) during a mission over Afghanistan. The UAV was captured by Iranian forces [1], which successfully extracted information, released a video obtained from the captured UAV, and subsequently constructed a copy of the RQ-170, the Saegheh [2]. As demonstrated by this episode, the theft of military UAVs causes significant damage, including the loss of secret data, such as collected information [3]. Consequently, it is indispensable to assure the security of the UAV [4–6].

Erasing data as needed can safeguard confidential information stored on remote devices. Generally, if a user sends an erasure request, the remote device receives it and performs the erasure. However, UAVs remotely performing a mission cannot always receive the signal [7–9]. As such, techniques are needed to completely erase the data stored in remote UAVs following a loss.

Recognition that sensitive information has been leaked (i.e., the erasure process has not been performed) is critical, and so confirmation of erasure must be performed in a verifiable manner (verifiable erasure). Verifying the result of the erasure operation as 1-bit response has low reliability [10].

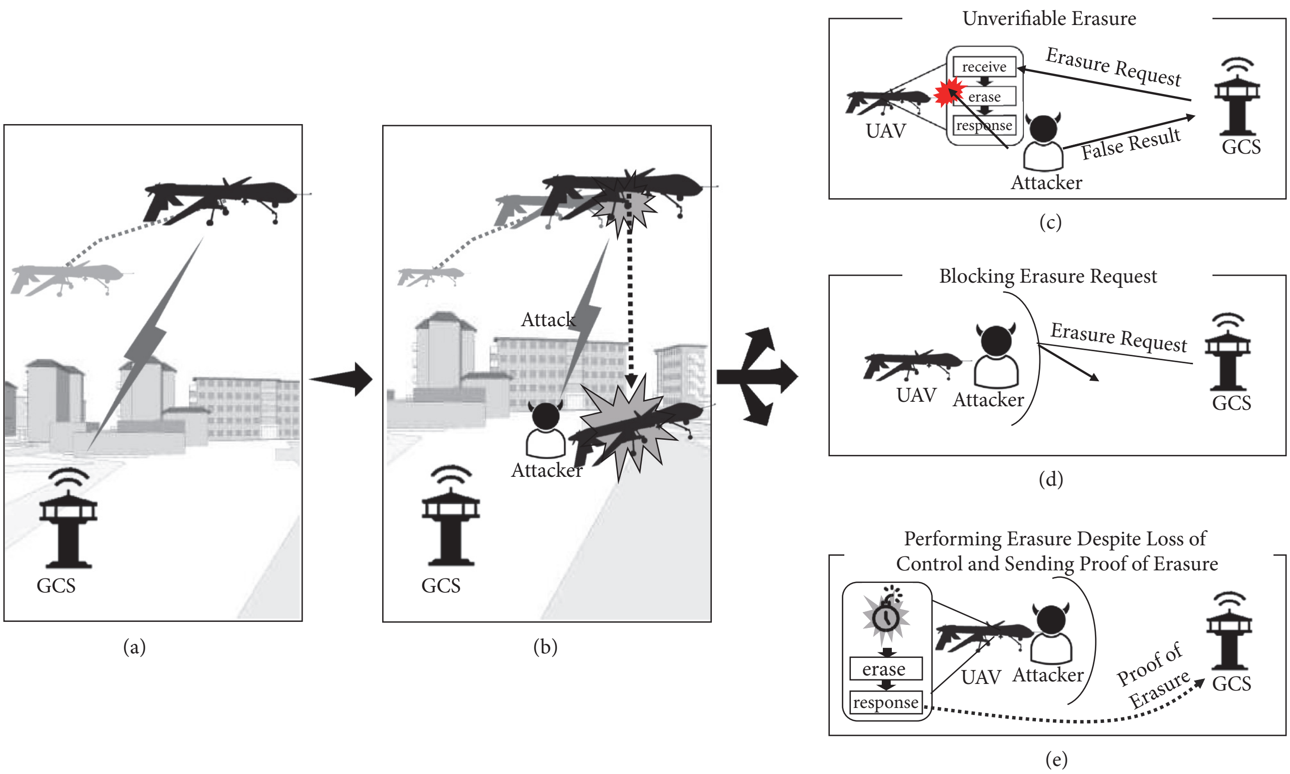


FIGURE 1: Three situations faced by unmanned aerial vehicles (UAVs) after attack or capture. (a) UAV on a flight mission; (b) UAVs performing a flight mission is captured by an attacker; (c) receiving unverifiable result of erasure; (d) blocked erasure requests by attackers; and (e) performing erasure operation and transmitting proof of erasure following capture.

Obvious proof of erasure must be presented, and there must be no mechanism by which malicious users can generate false evidence.

Figure 1 indicates three situations that UAVs may be faced with after attack or capture. In conventional remote erasure systems [11, 12], a Ground Control Station (GCS; that is, the control center that provides human control of UAVs) can erase data by sending an erasure signal to a lost UAV. However, when communication is cut, conventional remote erasure is not available. We devised counter-based erasure, which can erase data in an uncontrollable environment. This study is an extension of our previous work [13, 14], in which we focused on the system design of trustworthy remote erasure for UAVs. However, our objective here was to devise a mechanism to select random seed data using accumulated data and develop a method for trustworthy remote erasure. The devised scheme has a count value, which decrements value one by one and erases data when the counter value reaches zero. The GCS can retain stored data only through periodically sending the specific value to the UAV. If the GCS does not send the specific value to the remote UAV and the counter reaches zero, the UAV erases the data. In terms of verification, most existing mechanisms provide nothing or only a simple response (erase or not erase). Our remote erasure system provides relatively complex but verifiable proof of erasure through randomized data synchronization. Upon a process of erasure of data, the remote UAV continuously generates a proof of erasure until the UAV is completely stopped; this proof is repeatedly sent to the GCS. The GCS checks that the proof has been received and verifies that remote data are erased.

The remainder of this paper is organized as follows: Section 2 reviews the related works of verifiable remote data erasure. Section 3 describes the materials and gives details of our mechanism UAV-Undertaker. Section 4 describes two experiment results about overhead in terms of communication and computation. This paper ends in Section 5 with conclusion on our work.

2. Related Works

To resolve problem of erasure of remotely stored data with verifiability, the researchers have devised approaches which generate a provable value of erasure operation. We analyzed several of them and identified the limiting factors in applying a verifiable erasure for UAVs.

Perito and Tsudik [15] use a PoSE (Proof of Secure Erasure) for secure code updates on embedded devices by verifying the erasure of memory. A verifier sends verifier-selected randomness of the size of memory to a prover, and the prover's memory is filled with the received value. The prover returns the very same randomness written on the memory to the verifier. As both the verifier and the prover send a random value of memory size to each other, this

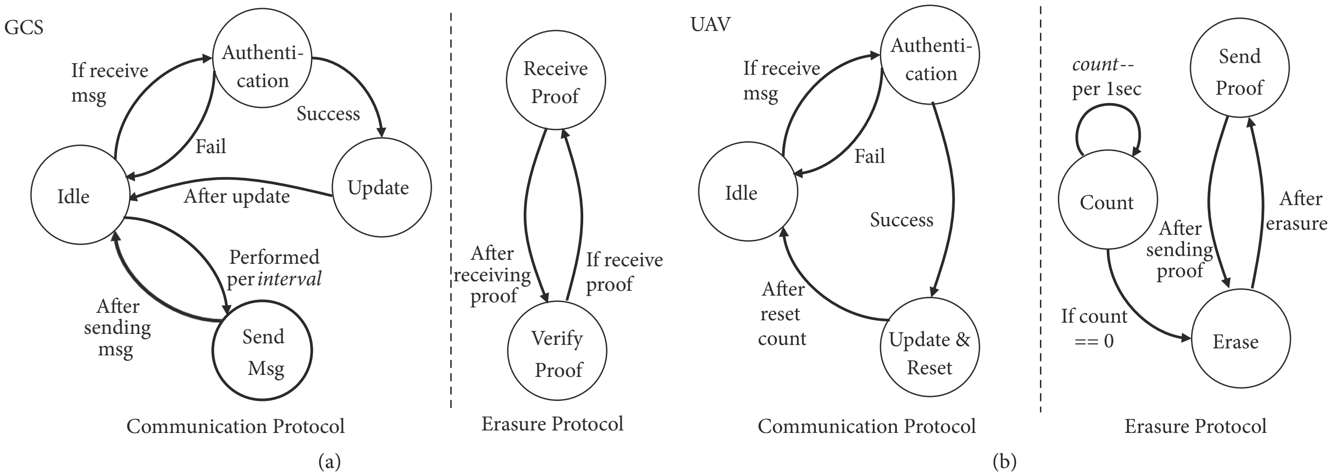


Figure 2: Overall process of the proposed remote erasure system. (a) Unmanned aerial vehicle (UAV)-Undertaker from the Ground Control Station (GCS) viewpoint; (b) UAV-Undertaker from the UAV viewpoint. If the count value becomes zero, the UAV erases data and repeatedly sends the proof of erasure to the user (i.e., the GCS).

protocol has a huge overhead concerning communication. In our system, as shown in Figure 9 of Section 4, when a small amount of data (i.e., α =2) is selected as a seed and transmitted, the instantaneous amount of transferred data is significantly lower than when live tracking all data (i.e., α =100). In other words, our system shares the burden of the required data transfer amount to verify the erasure operation. Therefore, our system can significantly reduce instantaneous amount of transferred data which is a limitation of the above study. Dziembowski et al. [16] proposed a scheme that reduces the communication complexity of PoSE. A verifier sends a seed to a prover. The prover performs a hash function using the seed and retains the sizable result value using all of the prover's memory. The calculated hash value can only be generated once because it uses a secret key stored in the memory, and so it can be proof of erasure. However, while this scheme reduces the communication overhead of PoSE, a huge calculative overhead arises. Ammar et al. [17] introduced SPEED, which guarantees erasure on embedded system. The protocol is implemented in isolated memory using the Trusted Software Module. A verifier's distance is measured by the Distance Bounding (DB) protocol, which establishes an upper bound on the physical distance between a verifier and a prover. The prover authenticates the verifier through the DB protocol and erases data if verification is passed successfully. To verify erasure, the prover sends the proof as the message authentication code value of the entire memory. However, the scheme is limited in terms of distance and so is not applicable for UAVs.

Our approach solves verifiable erasure through proof of erasure [15–18] that only the user can identify, as in the above studies. However, our approach creates a proof of erasure in such a way that the memory blocks each have a dependency on the block, which guarantees more robust evidence. Furthermore, in contrast to past studies, which did not consider losses in the communication environment, we aimed to achieve remote erasure of UAVs after hijack, loss, and/or disconnection. Also, the sensor system in cloud is similar to our system in that it collects data from remote devices and processes them to achieve specific goals. However, in the case of the cloud sensor system, the collected information is limited to the sensor data and is merely used for providing the monitoring to the user. In the case of our system, the collected information is the memory information of the remote device. It can be seen that there is a difference from the cloud sensor system in that it verifies whether or not the erasure operation is performed by utilizing this.

3. Proposed System Architecture

Our UAV-Undertaker consists of two parts: the communication protocol and verifiable erasure protocol. We confirm whether or not control of the UAV is lost by communicating with the UAV. Therefore, it is important to verify that a message originates from the stated sender (i.e., authentic communication) and has not been changed. This authentication role is implemented as the communication protocol using a hash chain mechanism [19]. We also implement the verifiable erasure protocol, which generates proof of erasure to verify that the erasure operation was really performed. We classify the data region according to the number of times the data will be written in order to increase the probative power of proof of erasure with just a small amount of data (thus, improving efficiency). The operation result value is created by accurately performing the implemented erasure operation. Therefore, a cryptographic accelerator, such as the TPM (Trusted Platform Module), must be used in order to make the results of the computation trustworthy.

Figure 2 shows the overall process of our approach from both the GCS and UAV viewpoints. From the GCS viewpoint, the GCS sends an authentication request message to the UAV at *interval*s and then waits for a message from the UAV. If the GCS receives a message from the UAV, it verifies that the message is authentic; if so, it updates the value to be

Table 1: Notations of the entity and messages.

Definition of the Entity Symbols	
GCS	Ground Control Station
UAV	Unmanned Aerial Vehicle
Denotation	
msg = $\alpha \parallel \beta$	Message contains two contexts (α and β)
msg = $\alpha \parallel (\beta)$	Message always contains α context, but β context is optionally contained.
Definition of the Message Symbols	
$K_{x,y}$	A symmetric key shared between x and y
$E\{K, B\}$	Encrypt B with key K
x	Value to compute with hash function
n	Number of hash function calculations
r	Retransmission bit
i	Authentication count
$h(x)$	Result of x in the hash function
Nonce	Generated random data against replay attacks
hl	Address of hot data modified after previous authentication
al	Address of accumulated data modified after previous authentication
hv	Set containing the addresses of hot data randomly selected and the values of those data
av	Set containing the addresses of accumulated data randomly selected and the values of those data
PoE	Proof of Erasure
Definition of the Greek Symbols	
α	Ratio of amount of randomly selected data in accumulated data region
β	Ratio of amount of randomly selected data in hot data region
γ	Number of re-authentication permits
δ	Initial value of the counter

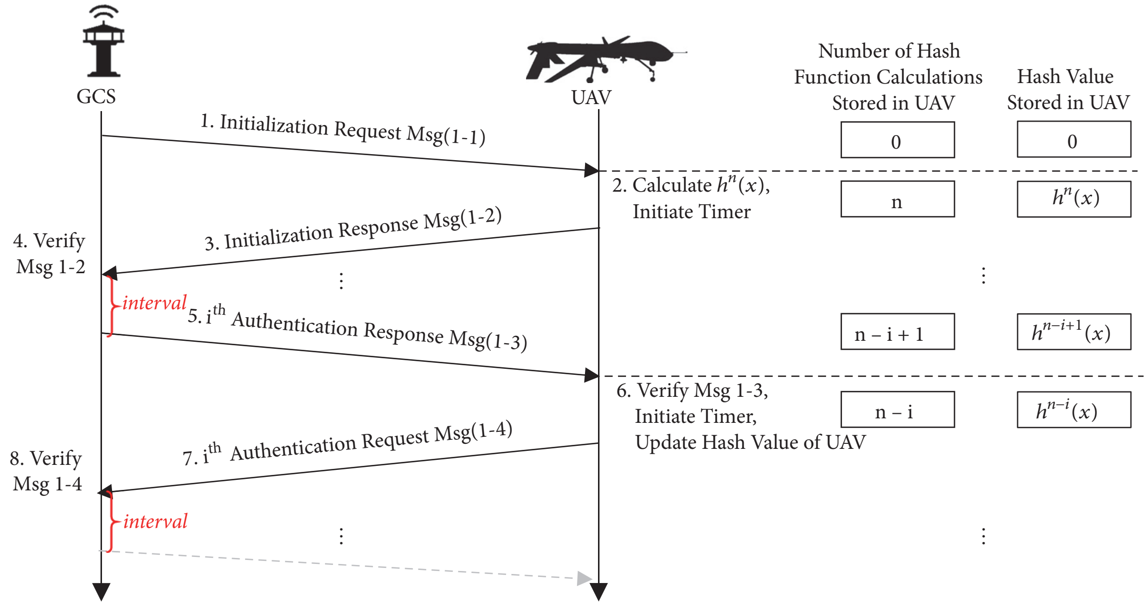
sent next. If communication with the UAV is cut off and the UAV performs the erasure operation and the GCS repeatedly receives the proof of erasure from the UAV and verifies it. From the UAV viewpoint, after decrementing the counter by one, the UAV waits for a message from the GCS (idle state). A UAV that has successfully authenticated a GCS update of the hash value for the next authentication, initializes the decreasing count value to δ (i.e., the initial value of the counter), and waits for the message to be received again (idle state). If the UAV does not receive a message from the GCS before the count value reaches a certain value, the UAV repeatedly erases stored data and repetitively sends proof of erasure through a low frequency.

3.1. Communication Protocol. To allow the sender of each message to be authenticated, hash chains and message encryption are used in the communication protocol. To encrypt a message, the UAV must share the symmetric key securely offline with the GCS before departure. Our approach is designed to execute the erasure protocol after γ times of requests for reauthentication, because the UAV may be confronted with a mere transient communication failure, where γ can be set freely according to the circumstances of mission. Table 1 defines the entity symbols and the message symbols.

Figure 3 shows the message flow when communication between the UAV and the GCS is performed without problems. First, the GCS encrypts the number of hash function calculations (n) and the value to compute with hash function (x) using a symmetric key already shared between the GCS and the UAV; it then sends an initialization request message (*Message 1-1*) containing that encrypted value to the UAV. When the value is received from the GCS, the UAV computes $h^n(x)$, encrypts it, and sends an initialization response message (*Message 1-2*) to the GCS. The GCS confirms that the UAV received the correct x and n. The GCS and the UAV, who have verified each other, start exchanging authentication messages using a hash chain. The GCS sends messages to the UAV at regular intervals. When the maximum count value is δ (seconds) and the number of reauthentication times is γ, the *interval* is calculated using (1). The constant "1" in (1) is added because GCS waits for the *interval* and then sends the next message. The constant "2" in (1) is multiplied to prevent the timer from being initialized during the maximum authentication latency that can occur on our system.

$$interval = \frac{\delta}{2(1+\gamma)} \quad (seconds) \tag{1}$$

After the UAV receives a message (*Message 1-3*) containing the hash value from the GCS, the UAV calculates the hash value of the received value and compares the value with the previously stored value. If two values are equal, the UAV initializes the count value to δ and updates the stored value with the hash value received from the GCS.

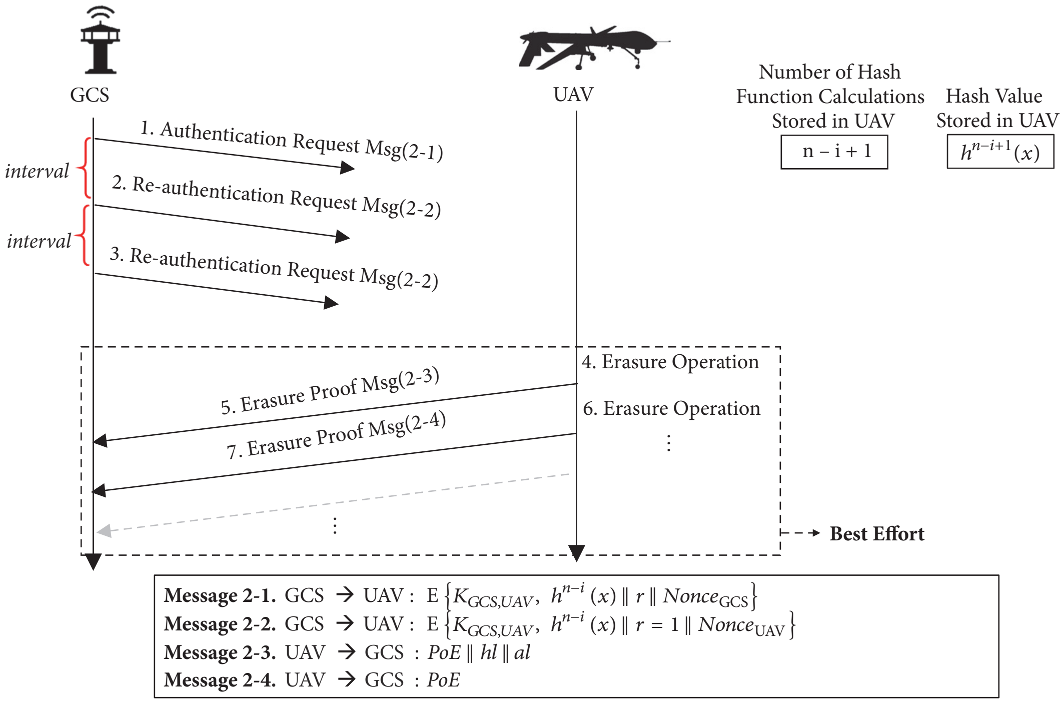

GCS
UAV
Number of Hash Function Calculations Stored in UAV
Hash Value Stored in UAV
1. Initialization Request Msg(1-1)
0
0
2. Calculate $h^n(x)$, Initiate Timer
n
$h^n(x)$
3. Initialization Response Msg(1-2)
4. Verify Msg 1-2
interval
5. i^{th} Authentication Response Msg(1-3)
n – i + 1
$h^{n-i+1}(x)$
6. Verify Msg 1-3, Initiate Timer, Update Hash Value of UAV
n – i
$h^{n-i}(x)$
7. i^{th} Authentication Request Msg(1-4)
8. Verify Msg 1-4
interval

Message 1-1. GCS → UAV : $E\{K_{GCS,UAV}, x \parallel n \parallel Nonce_{GCS}\}$
Message 1-2. UAV → GCS : $E\{K_{GCS,UAV}, h^n(x) \parallel Nonce_{GCS}\}$
Message 1-3. GCS → UAV : $E\{K_{GCS,UAV}, h^{n-i}(x) \parallel r \parallel Nonce_{GCS}\}$
Message 1-4. UAV → GCS : $E\{K_{GCS,UAV}, n-i \parallel Nonce_{UAV} \parallel hl \parallel al\} \parallel (hv \parallel av)$

Figure 3: Message flow depending on the communication environment. Messages 1-1–1-4 are those sent when communication between the unmanned aerial vehicle (UAV) and the Ground Control Station (GCS) is performed without problems.

After confirming that the message is sent from the GCS, the UAV sends a message (*Message 1-4*) containing the memory information along with the n − i to the GCS. *hv* and *av* (changed memory information) contained in *Messages 1-4* are not encrypted and transmitted. The reason is that, in our system, *hv* and *av* can include very large amounts of data and encrypting these data is expected to have a very high computational overhead. Even if the malicious user successfully acquires the unencrypted *hv* and *av*, the user cannot generate the false proof because the last hash value used in the erasure operation cannot be determined. These transactions are repeated *n* times if communication is good. When the number of authentication (*i*) exceeds *n* (i.e., the maximum number of hash function calculations), the GCS and the UAV reexchange the new *n* and *x* through *Message 1-1* and *Message 1-2*.

Figure 4 shows the message flow from when UAV could not receive the hash value from the GCS until the counter value reaches 0, at which point it initiates the erasure protocol. If the GCS sends an authentication request message (*Message 2-1*) to the UAV but does not receive the authentication response message from the UAV, the GCS waits the *interval* time and then sends again the message containing the same hash value. At this time, the *r* bit (i.e., the retransmission) is set to 1 and included in the message (*Message 2-2*). If the UAV faced a temporary communication failure, it will subsequently receive a reauthentication request message (*Message 2-2*) from the GCS; the UAV checks the *r* bit, recognizes resending, and confirms that the hash value stored in the UAV and received value are equal. If the two values are equal, the UAV returns value (n − i), which means the next hash value request. At that time, the UAV does not update the stored counter value or hash value. If the two values are not equal, the UAV calculates the hash value of the received value and compares it with the stored value. If the two values are equal, the UAV resets the counter value and updates the stored hash value. After that, the user and the UAV resume normal communication. However, if the next hash value is not received from the GCS and the value of the counter inserted in the UAV reaches 0, the UAV performs an erasure protocol and sends a message (*Message 2-3*) including proof of erasure and the memory information changed since the most recent successful authentication to the GCS through a certain frequency. In order for the GCS to verify that the UAV performed an erase operation, the GCS must know the changed memory information in the UAV. However, after the communication between the GCS and the UAV is lost, the GCS do not know information of data changed in the UAV. Thus, *Message 2-3* contains the changed memory information since the communication was disconnected. As our approach takes the best effort approach, the UAV repeatedly performs the erasure operation after transmitting the proof of erasure and continues to send the proof of erasure (*Message 2-4*) to the GCS. The number of times an erase operation is performed depends entirely on a given amount of time to the UAV after the erase operation has begun.


GCS
UAV
Number of Hash Function Calculations Stored in UAV
n − i + 1
Hash Value Stored in UAV
$h^{n-i+1}(x)$
interval
interval
1. Authentication Request Msg(2-1)
2. Re-authentication Request Msg(2-2)
3. Re-authentication Request Msg(2-2)
4. Erasure Operation
5. Erasure Proof Msg(2-3)
6. Erasure Operation
7. Erasure Proof Msg(2-4)
Best Effort

Message 2-1. GCS → UAV : $E\{K_{GCS,UAV},\ h^{n-i}(x) \| r \| Nonce_{GCS}\}$
Message 2-2. GCS → UAV : $E\{K_{GCS,UAV},\ h^{n-i}(x) \| r = 1 \| Nonce_{UAV}\}$
Message 2-3. UAV → GCS : $PoE \| hl \| al$
Message 2-4. UAV → GCS : PoE

FIGURE 4: Message flow depending on the communication environment. If the unmanned aerial vehicle (UAV) does not receive the hash value from the Ground Control Station (GCS) before the counter value reaches 0, the UAV proceeds with the erasure process and sends proof of erasure to the GCS.

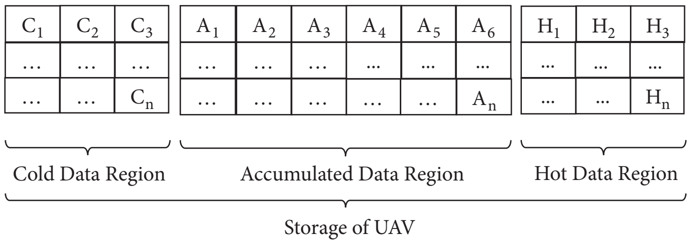


FIGURE 5: Unmanned aerial vehicle (UAV) storage divided into three regions: cold data region, accumulated data region, and hot data region.

3.2. Verifiable Erasure Protocol. For data erasure, our approach classifies the UAV storage into three data regions according to the expected number of writes of the data (Figure 5). The first region is a cold data region (i.e., data that will never be modified after the UAV departs), the second is the accumulated data region (i.e., data that will not be modified after it has been written; for example, video shot by the UAV), and the third is a hot data region (i.e., data that is actively modified). The mechanism assumes that the GCS knows the address values of the three data regions.

Our proposed proof of erasure generation method requires the live tracking of accumulated data and hot data (except for cold data that does not change because each data point has a dependency on each other in the proof of erasure). However, since tracking all the data causes considerable communication overhead, we randomly select data to act as seeds and transmit authentication messages containing just those data (Figures 6 and 7).

The selected data is managed by recording the address value and data type in the seed block table. In the accumulated data region, the number of seed data selects α% of the number of modified data blocks from time t_{n-1}, at which the previous seed block was selected to the current time t_n, at which the seed block selection occurs. Therefore, when selecting seed data, the seed is selected uniformly regardless of the time at which the data were generated. In the hot data region, the number of seed data selects *the hot data region size* $\times \beta$ % among modified data blocks from time t_{n-1}, at which the previous seed block was selected to the current time t_n, at which the seed block selection occurs. The value of α and β can be directly selected by the user. If the GCS and the UAV are in very good communication (high bandwidth) and there is no problem sending and receiving a lot of data; the GCS can increase the probative power of proof by setting the variable (α) high. Conversely, if the GCS and the UAV are in poor communication (low bandwidth) and cannot exchange

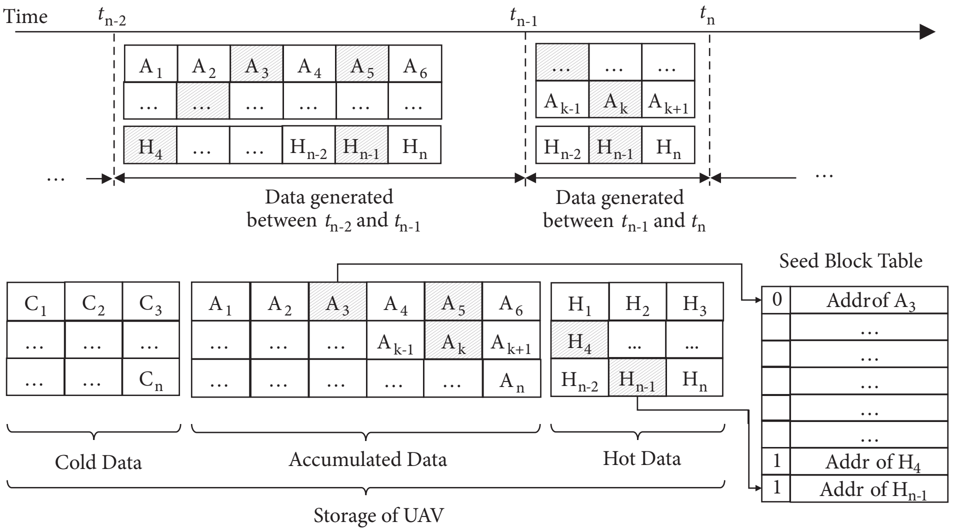

Time
t_{n-2}
t_{n-1}
t_n
A_1 A_2 A_3 A_4 A_5 A_6
H_4 … … H_{n-2} H_{n-1} H_n
A_{k-1} A_k A_{k+1}
Data generated between t_{n-2} and t_{n-1}
Data generated between t_{n-1} and t_n
Seed Block Table
0 Addrof A_3
1 Addr of H_4
1 Addr of H_{n-1}
C_1 C_2 C_3 C_n
A_n
H_1 H_2 H_3
Cold Data
Accumulated Data
Hot Data
Storage of UAV

Figure 6: Random selection of data from unmanned aerial vehicle (UAV) storage and the seed block table. The gray block of memory indicates data selected as seed blocks. The seed blocks are selected from data generated between time t_{n-1}, at which the previous seed block was selected, and time t_n at which the seed block selection occurs.

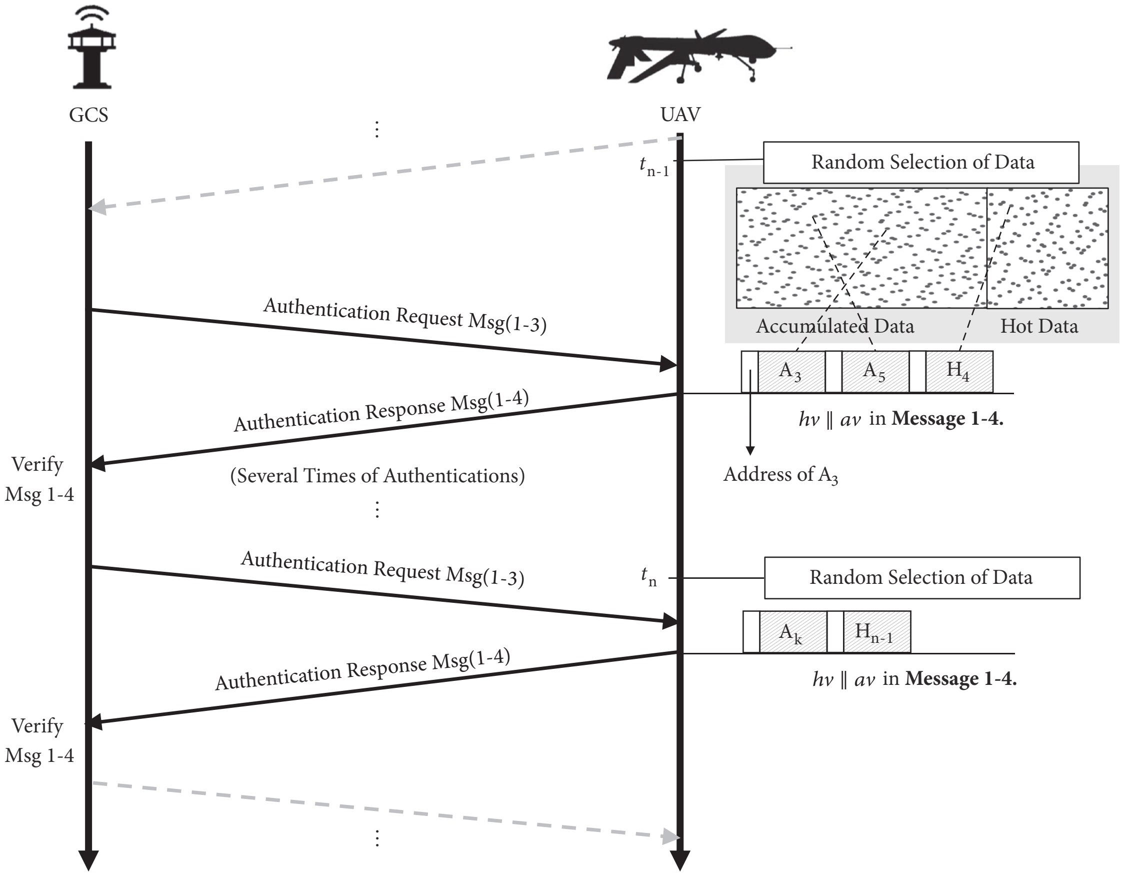

GCS
UAV
t_{n-1}
Random Selection of Data
Accumulated Data
Hot Data
A_3 A_5 H_4
$hv \parallel av$ in **Message 1-4.**
Address of A_3
Authentication Request Msg(1-3)
Authentication Response Msg(1-4)
Verify Msg 1-4
(Several Times of Authentications)
t_n
Random Selection of Data
A_k H_{n-1}
$hv \parallel av$ in **Message 1-4.**
Authentication Request Msg(1-3)
Authentication Response Msg(1-4)
Verify Msg 1-4

Figure 7: Process of sending randomly selected seed data to the Ground Control Station (GCS). The gray blocks represent randomly selected memory blocks, and t represents an arbitrary time.

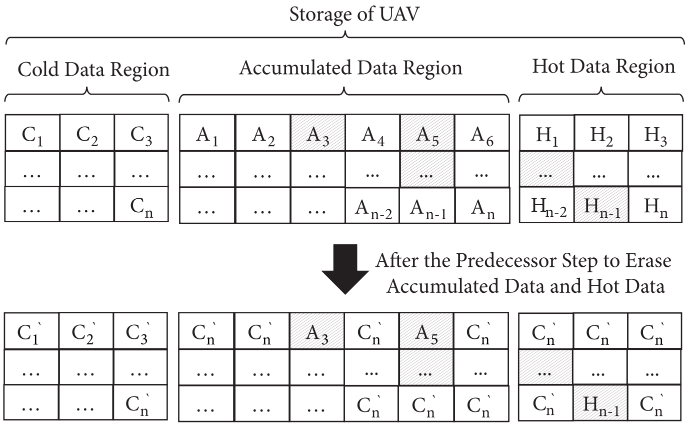

Storage of UAV
Cold Data Region
Accumulated Data Region
Hot Data Region
C1 C2 C3 … … … … … Cn
A1 A2 A3 A4 A5 A6 … … … ... ... … … … An-2 An-1 An
H1 H2 H3 ... ... Hn-2 Hn-1 Hn
After the Predecessor Step to Erase Accumulated Data and Hot Data
C1` C2` C3` … … … … … Cn`
Cn` Cn` A3 Cn` A5 Cn` … … … ... ... … … … Cn` Cn` Cn`
Cn` Cn` Cn` ... ... ... Cn` Hn-1 Cn`

Figure 8: Memory state of the unmanned aerial vehicle (UAV) after the predecessor step to erase accumulated data and hot data. In accumulated data and hot data regions, the remaining blocks, excluding the seed data blocks, are overwritten with $C_n{}'$, which is the result of the erasure operation of the cold data region.

large amounts of data, the GCS can lower the amount of communication data by setting the variable (α) very low. In other words, the GCS can set the variable according to the situation where the GCS use our system.

If the UAV counter reaches zero, the erasure protocol proceeds in the following order: erasure of cold data, erasure of accumulated data, and erasure of hot data. For erasure of cold data, the erasure operation is performed without any predecessor because the GCS already knows all of the values. The UAV overwrites the first block (C_1) of the cold data with the last hash value received from the GCS. The UAV performs an XOR operation of the C_2 block of the memory and C_1 block wiped with the last hash value received from the GCS and wipes the C_1 block with the calculated value. From now on, the wiped C_1 block is denoted by $C_1{}'$. Next, the C_2 block is wiped to the value obtained by the XOR operation of the $C_1{}'$ block and the C_2 block. The repeated wiping of memory proceeds in the same way (see (2)) until wiping the C_n block (i.e., the last block of cold data).

$$C_n{}' = C_n \oplus C_{n-1} \tag{2}$$

For accumulated data region and hot data, the GCS cannot know all of the stored values and so predecessors are required. The address value of the changed data after the last successful authentication time should be recorded in the main memory. The remaining blocks, excluding the seed data block, are overwritten with $C_n{}'$, which is the result of the erasure operation of the cold data. The bottom side of Figure 8 shows the memory state after this predecessor step. Then, the UAV performs the XOR operation with the A_2 block of accumulated data and the A_1 block of accumulated data and wipes the A_2 block with the result value. If confronted with the seed data (e.g., A_3) during the above operation, the UAV performs the XOR operation with the $A_2{}'$ block of accumulated data and the hash value of the A_3 block of accumulated data and wipes the A_3 block with the result value (see (3)). In this way, all data stored in the accumulated data region and the hot data region are erased. In other words, the seed data of hot data are also calculated from (4).

$$A_n{}' = h(A_n) \oplus A_{n-1} \tag{3}$$

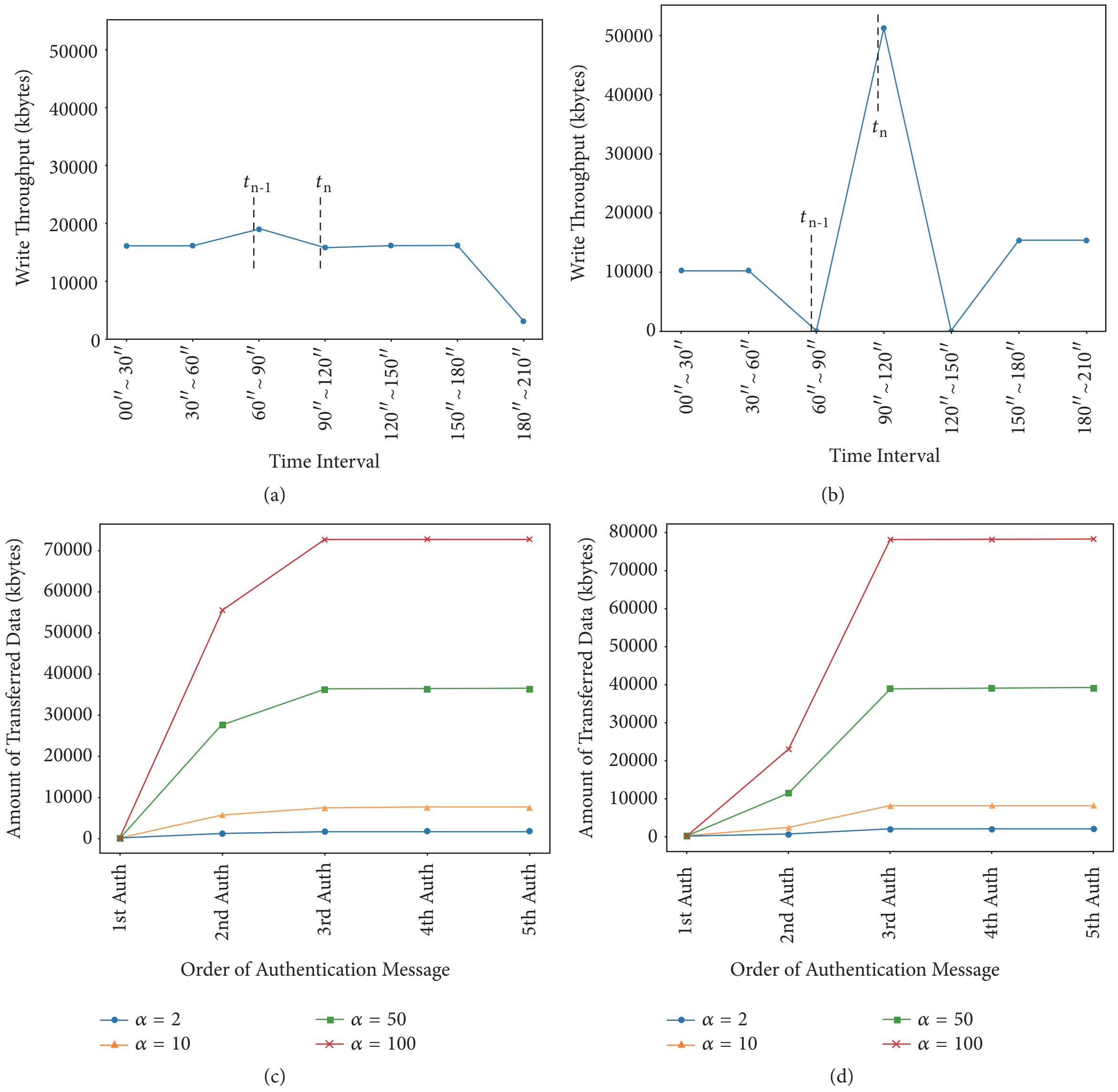
$$H_n{}' = h(H_n) \oplus H_{n-1} \tag{4}$$

The process above is defined as the first round. Starting from the second round, the erasure process proceeds without distinguishing the data region, unlike the first round. The UAV wipes $C_1{}'$ with the result of the XOR operation of the $C_1{}'$ block and $H_n{}'$ block. After that, the UAV wipes $C_2{}'$ with the result of the XOR operation of $C_1{}''$ block and $C_2{}'$. This process continues until the last block of hot data is overwritten, without any distinction of the data region, to complete the second round. To prevent forensic, we proceed through multiple rounds. We assume that the repeatedly wiped memory cannot be recovered, so it is impossible to extract information from the memory. After the second round process, as proof of erasure, the block $H_n{}''$ value is computed using a hash function, and the UAV sends that value (proof of erasure) to the GCS as Frequency Shift Keying (FSK) via the low frequency. The GCS who has received the proof of erasure from the UAV will be able to calculate the proof of erasure and confirm the validity of the message.

In the first part of the erasure process, the last hash value sent by the GCS is used as a seed in the erasure process to enhance security. The hash value used as the seed is an encrypted value, so an attacker cannot determine this value. Each value calculated by the above erasure process has a dependency on the previous memory data. Therefore, even if the attacker knows some blocks of memory, a false proof cannot be generated.

4. Experimental Results

To test performance of the devised protocol, we implemented an experimental environment using the *T2080* as the UAV flight computer [20]; the *T2080* has a 128 Mbytes NOR flash



(a)

(b)

(c)

(d)

Figure 9: Amount of data transferred according to the rate at which accumulated data are generated and *α*. (a) Constant amount of data written to the SD card over time interval; (b) random amount of data written to the SD card over time interval; (c) amount of transferred data when the data is generated as shown in Figure 9(a) on the unmanned aerial vehicle (UAV); and (d) amount of transferred data when the data is generated as shown in Figure 9(b) on the UAV.

memory, SD connector to interface, and SATA interface. Through this implementation, we conduct two experiments to investigate how practical the proposed scheme is: the communication overhead according to *α* and the erasure latency of the erasure process according to memory size. We conducted experiments on SD cards inserted in the *T2080*. The block size used in the XOR operation was determined based on how many 512-byte blocks were sequentially read at a time and the size of one block used in our experiment was 1Kbytes, which was determined by reading 512-byte blocks twice in succession.

4.1. Communication Overhead Experiment. This experiment measured the amount of data transferred according to the rate at which accumulated data are generated and *α*. The amount of transferred data in the experiment is the amount of data that is sent when the UAV sends an authentication response message to the user. To reproduce an environment similar to a real UAV's storage, we used a data generator called *iozone* [21] to write dummy data to the SD card. For 3 minutes and 30 seconds, a total of 100 Mbytes of accumulated data was generated through the data generator. Figures 9(a) and 9(b) show the amount of data written to the SD card over the time interval. In Figure 9(a), the data generator always writes a constant amount of data, but in Figure 9(b), the amount of data to write largely changes with time. Figure 9(c) shows the amount of transferred data when the data is generated as shown in Figure 9(a), and Figure 9(d) shows the amount of transferred data when the data is generated as shown in Figure 9(b). In both experiments, α was set to 2, 10, 50, and 100. Each experiment has two random selections of data, and the random selection of data occurred at the time indicated by $t_{\text{n-1}}$ and t_{n} in Figure 9. For comparison, the random selection of data occurred at the same time in both experiments. Authentication was performed every 45 seconds.

In our system, the amount of transferred data after a random selection of data occurs is greatly increased. As can be seen from the experimental results, the increased amount of transferred data is highly dependent on α. The amount of generated data also affects the amount of data transferred, but the amount of data transferred is very small if α is small. In other words, our system makes it possible to select α depending on the situation, enabling timer-based erasure operation and erasure verification even when the maximum transmission amount of the UAV is largely restricted. In addition, even if the amount of generated data suddenly increases, a sample of generated data is selected and transmitted, so it is possible to communicate more stably by sending less data during unstable communication situations.

4.2. Erasure Operation Overhead Experiment. In this experiment, we measured the erasure latency according to the memory size of the UAV, where the erasure latency was the time from when the embedded timer reaches 0 to when the proof of erasure is generated. Our approach divides the memory region into three regions according to the type of data and performs the erasure operation. So, we divided the memory used in the experiment into the following regions: 5% of the total memory for the cold data region, 85% for the accumulated data region, and 10% for the hot data region. Since we wanted the erasure latency to be affected only by memory size in this experiment, we equally set all the variables except for memory size. In this experiment, the values of α and β were set to 2. Figure 10 shows each number of the proof transmit for 1 Gbytes, 2 Gbytes, 4 Gbytes, and 16 Gbytes memory size when a limited time of 8 minutes is given to the UAV that initiated the erasure operation. The data transfer rate via FSK is recommended to be 345 bits per second [22]. Therefore, our experiment also sent 345 bits per second when transferring the proof of erasure. The amount of data changed after communication was disconnected and was assumed to be 100 Mbytes. The erasure latency recorded in the graph is the average value of erasure latency measured through three times of the erase operation.

As we might expect, the erasure latency of first round increased with the memory size. In the rest of the experiment, except for the 1 Gbytes scenario, the difference between the erasure latency of the first round execution and the erasure latency of the second round execution was large. This is because the number of hash operations increased as memory size increased. The first transmission of proof of erasure takes 9 seconds longer than the subsequent proof transmission, since it contains the address value of the changed data since communication was lost. As can be seen from the experimental results, if the UAV timer reaching zero, it has 8 minutes to perform the erasure operation 32 times at 1 Gbytes, 17 times at 2 Gbytes, 8 times at 4 Gbytes, and 2 times at 16 Gbytes. In other words, the UAV makes the best efforts in the given situation to erase the data and transfer the proof of erasure.

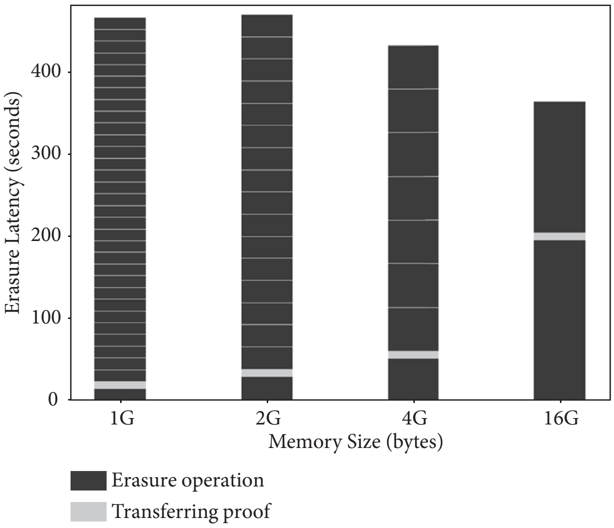


Figure 10: Experimental result for erasure latency according to the memory size of the unmanned aerial vehicle (UAV).

5. Conclusions

We have proposed a mechanism to provide proof of erasure operations after erasing data stored on UAVs, even if control of a remotely deployed UAV is lost. To do this, we used a countdown-based approach and hash chain to authenticate the sender of received messages and to trigger the erasure operation even after communication is lost. The accumulated data of the UAV is classified into new data types; the features of the data are actively utilized in the erasure operation, and the proof of erasure operation is transmitted so that the GCS can verify whether the erasure operation has actually been performed.

Our approach does not track all the data when generating proof of erasure, and so there is relatively little communication overhead; instead, we select seed data to generate proof of erasure. By allowing the amount of transferred data to generate proof through the value of α, timer-based erasure and erasure verification are possible even when the maximum amount of transmission is greatly limited. Furthermore, since a UAV that had lost control and started the erasure operation would not know what situation it was in, we used the best effort method to perform the erasure operation.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) [Grant no. NRF-2017R1C1B2003957] and by the Institute for Information & Communications Technology Promotion (IITP) of the Korea government (MSIT) [Grant no. 2019-0-00426 and no. 2018-0-00420].

References

[1] D. Cenciotti, *Iran Unveils New UCAV Modeled on Captured U.S. RQ-170 Stealth Drone*, The Aviationist, 2016, https://theaviationist.com/2016/10/02/iran-unveils-new-ucav-modeled-on-captured-u-s-rq-170-stealth-drone/.

[2] S. Shane and D. E. Sanger, *Drone Crash in Iran Reveals Secret U.S. Surveillance Effort*, New York Times, 2011, https://www.nytimes.com/2011/12/08/world/middleeast/drone-crash-in-iran-reveals-secret-us-surveillance-bid.html/.

[3] L. A. White, "The need for governmental secrecy: why the us government must be able to withhold information in the interest of national security," *Virginia Journal of International Law*, vol. 43, p. 1071, 2002.

[4] N. Uchida, S. Takeuchi, T. Ishida, and Y. Shibata, "Mobile traffic accident prevention system based on chronological changes of wireless signals and sensors," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 8, no. 3, pp. 57–66, 2017.

[5] C. Gritti, M. Önen, R. Molva, W. Susilo, and T. Plantard, "Device identification and personal data attestation in networks," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 9, no. 4, pp. 1–25, 2018.

[6] N. Uchida, K. Ito, T. Ishida, and Y. Shibata, "Adaptive array antenna control methods with delay tolerant networking for the winter road surveillance system," *Journal of Internet Services and Information Security (JISIS)*, vol. 7, no. 1, pp. 2–13, 2017.

[7] F. Arena, P. Giovanni, and M. Collotta, "A survey on driverless vehicles: from their diffusion to security features," *Journal of Internet Services and Information Security (JISIS)*, vol. 8, no. 3, pp. 1–19, 2018.

[8] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[9] A. Rawnsley, "Iran's alleged drone hack: tough, but possible," *Wired*, 2011, https://www.wired.com/2011/12/iran-drone-hack-gps/.

[10] F. Hao, D. Clarke, and A. F. Zorzo, "Deleting secret data with public verifiability," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, pp. 617–629, 2016.

[11] M. D. Leom, K. Kwang, R. Choo, and R. Hunt, "Remote wiping and secure deletion on mobile devices: a review," *Journal of Forensic Sciences*, vol. 61, no. 6, pp. 1473–1492, 2016.

[12] X. Yu, Z. Wang, K. Sun, W. T. Zhu, N. Gao, and J. Jing, "Remotely wiping sensitive data on stolen smartphones," in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, ASIA (CCS)*, pp. 537–542, ACM, Japan, 2014.

[13] S. Kim, T.-Y. Youn, D. Choi, and K.-W. Park, "Securely controllable and trustworthy remote erasure on embedded computing system for unmanned aerial vehicle," in *Proceedings of the 3rd International Symposium on Mobile Internet Security (MobiSec)*, vol. article 8, pp. 1–9, Cebu, Philippines, 2018.

[14] S. Kim, T.-Y. Youn, D. Choi, and K.-W. Park, "Securely controllable and trustworthy remote erasure on embedded computing system for unmanned aerial vehicle," in *Proceedings of the Research Briefs on Information & Communication Technology Evolution (ReBICTE)*, vol. 4, article 3, 2018.

[15] D. Perito and G. Tsudik, "Secure code update for embedded devices via proofs of secure erasure," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 643–662, Springer, 2010.

[16] S. Dziembowski, T. Kazana, and D. Wichs, "One-time computable self-erasing functions," in *Theory of Cryptography*, vol. 6597 of *Lecture Notes in Computer Science*, pp. 125–143, Springer, Heidelberg, Germany, 2011.

[17] M. Ammar, B. Crispo, W. Daniels, and D. Hughes, "Speed: secure provable erasure for class-1 iot devices," in *Proceedings of the 8th ACM Conference on Data and Application Security and Privacy (CODASPY)*, pp. 111–118, 2018.

[18] G. O. Karame and W. Li, "Secure erasure and code update in legacy sensors," in *Proceedings of the International Conference on Trust and Trustworthy Computing*, vol. 9229, pp. 283–299, Springer International Publishing, 2015.

[19] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[20] M. Slonosky, "Trusted boot in COTS computing," *Military Embedded Systems*, 2015, http://mil-embedded.com/articles/trusted-boot-cots-computing/.

[21] W. D. Norcott, "Iozone filesystem benchmark," http://www.iozone.org/docs/IOzone_msword_98.pdf.

[22] L. Deshotels, "Inaudible sound as a covert channel in mobile devices," *WOOT*, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 7840917, 11 pages
https://doi.org/10.1155/2019/7840917

Research Article

Authorized Client-Side Deduplication Using CP-ABE in Cloud Storage

Taek-Young Youn,[1] **Nam-Su Jho**,[1] **Kyung Hyune Rhee**,[2] **and Sang Uk Shin**[2]

[1] *Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea*
[2] *Department of IT Convergence and Application Eng., Pukyong National University, Busan 48513, Republic of Korea*

Correspondence should be addressed to Sang Uk Shin; shinsu@pknu.ac.kr

Received 17 January 2019; Revised 11 March 2019; Accepted 15 April 2019; Published 15 May 2019

Academic Editor: Ilsun You

Since deduplication inevitably implies data sharing, control over access permissions in an encrypted deduplication storage is more important than a traditional encrypted storage. Therefore, in terms of flexibility, data deduplication should be combined with data access control techniques. In this paper, we propose an authorized deduplication scheme using CP-ABE to solve this problem. The proposed scheme provides client-side deduplication while providing confidentiality through client-side encryption to prevent exposure of users' sensitive data on untrusted cloud servers. Also, unlike existing convergent encryption schemes, it provides authorized convergent encryption by using CP-ABE to allow only authorized users to access critical data. The proposed authorized deduplication scheme provides an adequate trade-off between storage space efficiency and security in cloud environment and is very suitable for the hybrid cloud model considering both the data security and the storage efficiency in a business environment.

1. Introduction

Cloud computing greatly facilitates data providers who want to outsource their data to the cloud without disclosing their sensitive data to external parties and would like users with certain credentials to be able to access the data [1]. This requires that the data be stored in an encrypted form that supports access control policies so that no one but a user with a particular type of attributes (privilege information) can decrypt the encrypted data. On the other hand, the amount of data stored in the storage of cloud storage providers (CSPs) is growing very rapidly, especially at the age of big data. Therefore, one of the important issues of the CSP is how to efficiently manage the ever-increasing data. One of the important techniques to solve this problem is deduplication. Deduplication is a special data compression technique to remove redundant copies of repeated data and can be used to effectively reduce data storage space and communication overhead.

However, in the corporate environment, employees have different permissions to information depending on their job or department. That is, different department employees have different access rights to information according to the various access control systems used [2]. Each employee can only access files that correspond to its privileges. If the deduplication technique does not check file privileges, it would violate file access rights, which would bring some security problems. Since deduplication inevitably implies data sharing, control over access permissions in encrypted deduplication storage is more important than a traditional encrypted storage. Therefore, in terms of flexibility, data deduplication should be combined with data access control techniques. That is, even if encrypted, the same data must be stored only once in the cloud, and it must be able to control access by other users based on the policy of the data owner. Existing secure deduplication techniques do not support access policy based encryption. In the current cloud computing environments, access policy-based encryption (typically ABE (Attribute Based Encryption) scheme) and secure deduplication are widely adopted separately. However, the standard ABE technique does not support secure deduplication, a technique that helps save storage space and network bandwidth by removing redundant copies of encrypted data in the cloud. Therefore, a design of a cloud storage system supporting both of these properties is required.

In this paper, we propose an authorized deduplication scheme using CP-ABE (Ciphertext-Policy Attribute-Based Encryption) to solve this problem. The proposed scheme provides client-side deduplication while providing confidentiality through client-side encryption to prevent exposure of users' sensitive data on untrusted cloud servers. Also, unlike existing convergent encryption schemes, it provides authorized convergent encryption by using CP-ABE to allow only authorized users to access critical data. The proposed scheme satisfies security requirements and has advantages over existing schemes. The proposed authorized deduplication scheme provides an adequate trade-off between storage space efficiency and security in cloud environment and is very suitable for the hybrid cloud model considering both the data security and the storage efficiency in a business environment.

This paper is organized as follows. Section 2 describes related works. In Section 3, we propose and analyse an authorized client-side deduplication using CP-ABE. And Section 4 describes the optimization and discussions. Finally, Section 5 is the conclusions.

2. Related Works

2.1. Secure Deduplication. Deduplication can be classified into client-side deduplication and server-side deduplication, and client-side deduplication is advantageous from the viewpoint of the efficient use of bandwidth. Therefore, many studies on client-side deduplication have been made. Despite the many advantages of deduplication technique, deduplication technique for the important data has caused several new security problems. In particular, securing the confidentiality of outsourced data is a very important issue, and it may be considered to perform an encryption operation before outsourcing. When a conventional encryption is used, each user has a different key, so that different ciphertexts are computed for the same plaintext. Therefore, deduplication cannot be achieved in this case. To solve this problem, Douceur et al. proposed a convergent encryption using a hash value of a plaintext as an encryption key [3]. Here, the encryption scheme $E()$ is a deterministic algorithm, and a convergent key K depends only on the input data file F. In most cases, however, a convergent encryption is vulnerable to offline brute-force attacks because the plaintext space of a given ciphertext C is not large enough (messages are often predictable) [4]. An attacker can perform the encryption operation for all possible plaintexts at the offline stage to find the corresponding plaintext information. To solve this problem, Bellare et al. proposed a technique called DupLESS that generates a convergent key through the interaction with a key server [5]. Duan et al. proposed a technique for generating a convergent key with the help of other users without a key server [6]. Miao et al. also proposed a method by modifying a single key server in DupLESS as a group of key servers [7].

In [8], Li et al. first proposed a deduplication scheme that applies the privilege information. In this scheme, the privilege information is applied when calculating an authentication tag of the file for deduplication. The authentication tag is generated by the private cloud functioning as the authorized server, and the private cloud possesses the privilege private key corresponding to the user's privileges, which are not distributed to the user. Therefore, users with the same privilege information can generate the authentication tag identifying deduplication, so deduplication is possible. Shin et al. applied an access privilege to compute a convergent key [9]. Privilege information is applied in RSA blind signature based oblivious PRF protocol in order to allow only the authorized user to access to data.

2.2. CP-ABE. Sahai and Waters introduced Attribute Based Encryption (ABE) which is the public key cryptography of one-to-many algorithm and encrypts the data based on the set of attributes [10]. There are two kinds of ABE schemes: Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE) [11]. In KP-ABE schemes, the ciphertext is associated with a set of attributes while the user's private key is generated based on his corresponding access policy, while in CP-ABE schemes, a user's private key is associated with a set of attributes, and ciphertext is encrypted under a specified access structure [12]. A user is able to decrypt a ciphertext only if the attributes associated with the ciphertext/private key satisfy the access policy related to his private key/ciphertext. Most ABE schemes [11, 12] are constructed based on bilinear pairing. We also use CP-ABE scheme based on bilinear pairing in [12].

In ABE scheme, attributes can be defined with different types of privilege properties. An access structure $\wp$ will contain some authorized sets of attributes. So an access policy can be defined as an access structure. An access structure can be represented by an access tree [11, 12]. In this paper, we use the definitions of an access structure and an access policy in [11, 12], and the identities of parties may be replaced by the attributes.

3. Authorized Client-Side Deduplication Using CP-ABE

In this section, we propose an authorized client-side deduplication scheme based on the CP-ABE. The proposed scheme provides the confidentiality through the client-side encryption to prevent the exposure of users' sensitive data on untrusted cloud servers. It also enables deduplication of the encrypted data through the convergent encryption, thereby saving the storage space in the cloud server. Unlike existing convergent encryption schemes, it provides an authorized convergent encryption based on CP-ABE to allow only authorized users to access critical data.

3.1. The System and Threat Model. The proposed scheme consists of User (U), Authorization Server (AS), and Cloud Service Provider (CSP), as shown in Figure 1. Here, the AS can be regarded as a private cloud and the CSP can be considered as a public cloud. The AS is an entity that helps a user to securely use a CSP. Also, the AS generates and manages to the user's privilege secret key corresponding to the access privilege and computes an authorized convergent key for a file by applying the privilege through the interaction with a user. The *CSP* provides data storage services to users, and deduplication technology is applied to save storage space and cost.

The proposed scheme consists of the system setup process, the authorized convergent key generation process, and the authorized deduplication process. In the system setup process, the system parameters are generated and the key pair of each entity is securely generated. The authorized convergent key generation process performs an interaction for the CP-ABE-based convergent key distribution between the user and the authorization server. Finally, the authorized deduplication process supports the client-side deduplication of the encrypted data using the generated convergent key, and the proofs of ownership (PoW) [13] protocol for verifying the file retention of the user are performed in this process (a concrete PoW protocol is not discussed in this paper, but a PoW protocol such as Merkle tree-based scheme [13] can be used).

We assume that the *AS* assumes an *honest-but-curious* trust model and the *AS* and the *CSP* do not collude. It is assumed that the user U can act maliciously. We consider that the *CSP* may act maliciously due to insider/outsider attacks, software/hardware malfunctions, intentional saving of computational resources, etc. [14]. According to this assumption, we consider the following types of adversaries:

(i) Outside adversary: it tries to obtain sensitive information from the cloud storage or tries to access the file beyond its access privileges.

(ii) Insider adversary: it can access the cloud storage easily and try to get information out of the user's encrypted data or file tags.

We consider the following security requirements that must be satisfied according to the threat model:

(i) The confidentiality of data stored in cloud: except for the information about duplication, no information about the outsourced data is revealed to an adversarial party.

(ii) The unforgeability of the file tag: a user without appropriate privileges should be prevented from generating file tags.

(iii) Secure deduplication: secure deduplication is supported without revealing any information except for the information about duplication.

3.2. The Proposed Scheme. As shown in Figure 2, the proposed scheme consists of the system setup phase, the authorized convergent key generation phase, and the file transfer phase. In the system setup phase, the user and the AS set their own keys. The authorized convergent key generation phase generates a user's convergent key for a file by reflecting the user's privilege information. The authorized file transfer phase contains the file upload and the file download, and the file upload is composed of the first upload module and the deduplication module depending on whether or not the files are duplicated. The detailed protocols are as follows.

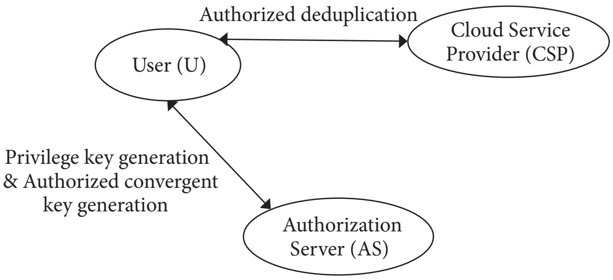

Authorized deduplication
User (U)
Cloud Service Provider (CSP)
Privilege key generation & Authorized convergent key generation
Authorization Server (AS)

Figure 1: System model.

3.2.1. The System Setup Phase. Users and the *AS* set their own keys by the security parameter λ. First, the *AS* selects the following as the system-wide public parameters:

(i) A cyclic group G of a prime order q generated by g

(ii) A bilinear paring $e : G \times G \longrightarrow G_T$ (a group G_T of order q)

(iii) A hash function $H : \{0,1\}^* \longrightarrow G$

(iv) A universe of privileges $\wp = \{p_1, p_2, \cdots, p_n\}$

The *AS* selects values $a, b \in \mathbb{Z}_q$ randomly. Then a master secret key (MSK) of the *AS* is (g^a, b) and a public key (PK) is $(g, g^b, e(g,g)^a)$. In addition, the *AS* generates a private key and a public key for the signing as follows:

(i) Private key for the signing, $x \xleftarrow[R]{} \mathbb{Z}_q$

(ii) Public key for the signing, $y = g^x$

It is assumed that the public parameters and the public key information are securely distributed throughout the system.

A user U obtains the privilege secret key SK_U corresponding to his access privilege $A_U = \{p_i\}$ $(A_U \subseteq \wp)$ from the *AS* over the secure channel. To do this, the user U sends (ID_U, A_U) to the *AS* to request the key generation. After choosing $t \in_R \mathbb{Z}_q$ and $t_j \in_R \mathbb{Z}_q$ for each attribute $j \in A_U$, the *AS* constructs the user's privilege secret key $SK_U = (D = g^{(a+t)/b}, \{D_j = g^t \bullet H(p_i)^{t_j}, g^{t_j}\}_{\forall j \in A_U})$ to securely transmit it to the user U (see [12] for details on the privilege secret key generation for CP-ABE scheme). The user U securely stores the privilege secret key SK_U. The user also generates a private key and a public key pair (*spk*, *ssk*) for a digital signature scheme such as DSA [15].

3.2.2. The Authorized Convergent Key Generation Phase. Before requesting the file upload, the user must perform the authorized convergent key generation phase. The user U generates a convergent key CK for encrypting and uploading the file F to the CSP as follows. This process modifies the RSA oblivious PRF-based convergent key generation method of [5] and generates a convergent key reflecting the user's privilege information by applying the CP-ABE scheme (see [12] for details on the encryption and decryption process for CP-ABE scheme). Figure 3 shows the authorized convergent key generation process.

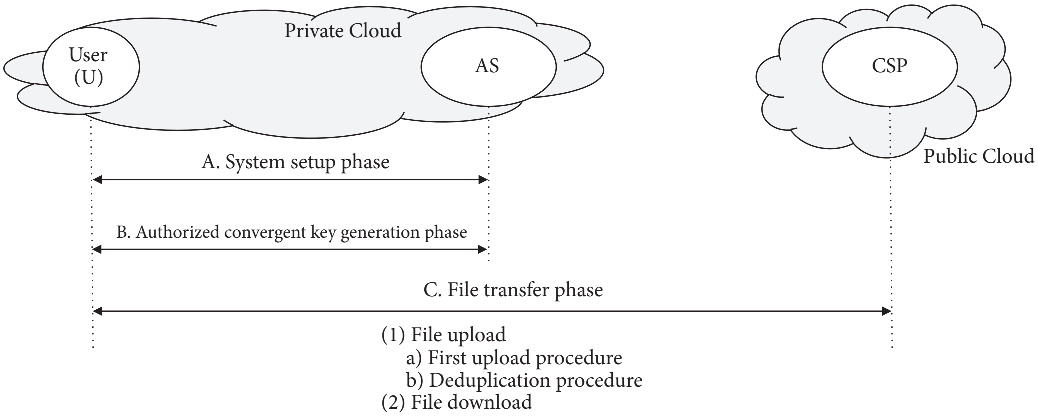


Figure 2: The proposed scheme.

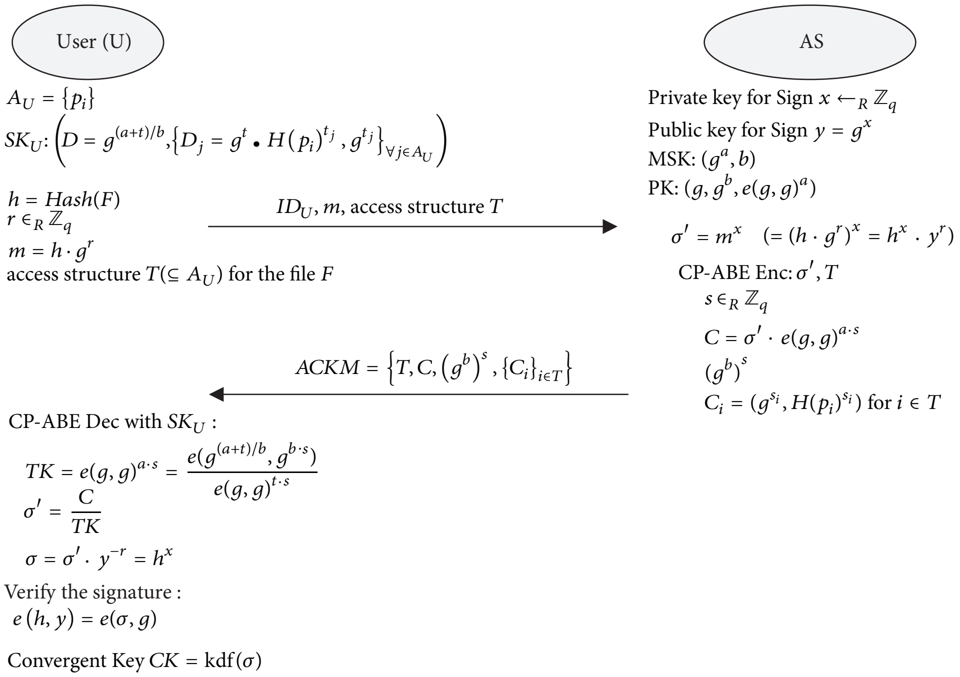

User (U)
AS
$A_U = \{p_i\}$
SK_U: $\left(D = g^{(a+t)/b}, \{D_j = g^t \bullet H(p_i)^{t_j}, g^{t_j}\}_{\forall j \in A_U}\right)$
$h = Hash(F)$
$r \in_R \mathbb{Z}_q$
$m = h \cdot g^r$
access structure $T (\subseteq A_U)$ for the file F
ID_U, m, access structure T
Private key for Sign $x \leftarrow_R \mathbb{Z}_q$
Public key for Sign $y = g^x$
MSK: (g^a, b)
PK: $(g, g^b, e(g,g)^a)$
$\sigma' = m^x \quad (= (h \cdot g^r)^x = h^x \cdot y^r)$
CP-ABE Enc: σ', T
$s \in_R \mathbb{Z}_q$
$C = \sigma' \cdot e(g,g)^{a \cdot s}$
$(g^b)^s$
$C_i = (g^{s_i}, H(p_i)^{s_i})$ for $i \in T$
$ACKM = \{T, C, (g^b)^s, \{C_i\}_{i \in T}\}$
CP-ABE Dec with SK_U :
$TK = e(g,g)^{a \cdot s} = \frac{e(g^{(a+t)/b}, g^{b \cdot s})}{e(g,g)^{t \cdot s}}$
$\sigma' = \frac{C}{TK}$
$\sigma = \sigma' \cdot y^{-r} = h^x$
Verify the signature :
$e(h, y) = e(\sigma, g)$
Convergent Key $CK = \text{kdf}(\sigma)$

Figure 3: The authorized convergent key generation process.

(i) A user U with access privileges $A_U = \{p_i\}$ constructs a request message for a convergent key generation for file F as follows and sends it to the AS:

(a) compute $h = Hash(F)$, where $Hash()$ is a cryptographically secure hash function, $Hash : \{0,1\}^* \longrightarrow \mathbb{Z}_q$.

(b) select $r \in_R \mathbb{Z}_q$

(c) compute $m = h \cdot g^r$

(d) construct an access structure T $(\subseteq A_U)$ for the file F (an access structure T is usually expressed in a tree form; see [12] for details) and send the request message for the convergent key generation to the AS.

(ii) The AS responds to the user U by constructing $ACKM$ (Authorized Convergent Key Material) as follows:

(a) compute $\sigma' = m^x$ $(= (h \cdot g^r)^x = h^x \cdot y^r)$

(b) perform $CP\text{-}ABE\ Enc(\sigma', T)$ with PK:

(1) select $s \in_R \mathbb{Z}_q$ and compute $(g^b)^s$ (where s is set to the value corresponding to the root node of the tree access structure T)

(2) compute $C = \sigma' \bullet e(g,g)^{a \bullet s}$

(3) output $C_i = (g^{s_i}, H(p_i)^{s_i})$ for the leaf node $i \in T$ (where s_i is a share of s corresponding to the leaf node i of the tree access structure T, see [12] for details)

(c) send $ACKM = \{T, C, (g^b)^s, \{C_i\}_{i \in T}\}$ to the user U.

(iii) The user U uses $ACKM$ to generate a convergent key CK and the file tag Tag_F.

(a) perform $\mathit{CP\text{-}ABE\ Dec}(SK_U, ACKM)$:

(1) compute $TK = e(g,g)^{a \bullet s} = e(g^{(a+t)/b}, g^{b \bullet s})/e(g,g)^{t \bullet s}$

$$\left(= \frac{e(g,g)^{(a+t)/b \bullet (b \bullet s)}}{e(g,g)^{t \bullet s}} = \frac{e(g,g)^{(a+t) \bullet s}}{e(g,g)^{t \bullet s}} = \frac{e(g,g)^{a \bullet s + t \bullet s}}{e(g,g)^{t \bullet s}} = \frac{e(g,g)^{a \bullet s} \bullet e(g,g)^{t \bullet s}}{e(g,g)^{t \bullet s}} \right) \quad (1)$$

where $e(g,g)^{t \bullet s} = \prod_{i \in T}\{e(g^t \bullet H(p_i)^{t_i}, g^{s_i}) / e(g^{t_i}, H(p_i)^{s_i})\} = \prod_{i \in T}\{e(g,g)^{t \bullet s_i}\}$ (for details of the computation, see [12])

(2) compute $\sigma' = C/TK$

(3) compute $\sigma = \sigma' \bullet y^{-r} = h^x$

(b) verify that the signature of the AS is correct: $e(h,y) = e(\sigma, g)$

(c) if the signature verification passes, the convergent key $CK = \mathrm{kdf}(\sigma)$ and the file tag $Tag_F = Hash(CK)$ are computed (where the function kdf() is a cryptographically secure key derivation function).

3.2.3. The File Transfer Phase. Suppose that a user $U1$ requests the CSP to upload a file F. The CSP performs duplication check on the file F and approves the upload of the encrypted file if there is no duplication. If there is duplication (that is, if the same file is already stored), then perform the deduplication process. At this time, the PoW process is performed to confirm whether the user holds the file actually. If the verification is passed, the owner of the file stored in the storage is assigned to the user $U1$ without uploading the actual file.

(1) File Upload (see Figure 4)

(i) Note that the user $U1$ already has $\{Tag_F, r, TK, CK\}$ for the file F through the authorized convergent key generation phase.

(ii) The user $U1$ transmits the upload request message $\{ID_{U1}, Tag_F, T\}$ for the file F to the CSP using the file tag Tag_F and the access structure T.

(iii) The CSP uses the tag Tag_F and the access structure T to check whether a duplicate file is stored in the storage. If there is no duplication, perform the "First Upload" module. If the duplicate file exists, perform the "Deduplication" module.

(a) First Upload module (see Figure 5)

(i) The CSP requests the file transfer to the user $U1$.

(ii) The user $U1$ computes the ciphertext $CT = Enc(CK, F)$ by encrypting the file F using the convergent key CK, where $Enc()$ is an authenticated encryption mechanism such as AES in CTR mode [16]. Also, $U1$ computes $vTK = Hash(TK, CK)$ and $sig = Sign_{ssk}(CT, ACKM, y^r, vTK\ )$, where $Sign_{ssk}()$ is a signing algorithm using the user's private key ssk. Then $U1$ sends $\{CT, ACKM, y^r, vTK, sig\}$ to the CSP. The user $U1$ stores the file tag Tag_F and deletes F, CK, $ACKM$, and so on.

(iii) The CSP verifies the signature sig using the user's spk. If valid, store the file record $\{Tag_F, CT, ACKM, y^r, vTK, ID_{U1}\}$.

(b) Deduplication module (see Figure 6)

(i) If the file F already exists, the CSP has the file record $\{Tag_F, CT, ACKM, y^r, vTK, ID_{U0}\}$ from the old user $U0$.

(ii) Note that $U1$ has $\{Tag_F, r', TK', CK\}$ for the file F through the authorized convergent key generation phase such that $\{r', TK'\} \neq \{r, TK\}$.

(iii) For the deduplication procedure, the CSP first performs PoW protocol with $U1$ for the ciphertext $CT = Enc(CK, F)$. Within this protocol, the CSP sends vTK, $ACKM$, and y^r to $U1$. If the user passes PoW protocol, the CSP assigns the file reference to the user $U1$ and adds ID_{U1} to the file record.

(iv) Within the PoW protocol, the user $U1$ recovers TK using $ACKM$ and verifies his own access privileges as

$$CK = \mathrm{kdf}\left(\frac{C}{TK} \bullet \frac{1}{y^r}\right) \quad (2)$$

and $vTK == Hash(TK, CK)$.

If the verification passes, $U1$ keeps the file tag Tag_F and deletes all other data from its own storage.

(2) File Download (see Figure 7)

(i) The user $U1$ has attributes $A_{U1} = \{p_i\}$ and its corresponding privilege secret key $SK_{U1} = (D = g^{(a+t)/b}, \{D_j = g^t \bullet H(p_i)^{t_j}, g^{t_j}\}_{\forall j \in A_{U1}})$.

(ii) $U1$ sends the File download request message with $\{ID_{U1}, Tag_F\}$ to the CSP.

(iii) The CSP finds Tag_F in the file records and load it. The CSP checks whether ID_{U1} is in the user lists of the record. If exists, the CSP sends $\{CT, ACKM, y^r, vTK\}$ to the user $U1$.

(iv) $U1$ performs $\mathit{CP\text{-}ABE\ Dec}$ on $ACKM$ with SK_{U1} as follows and obtains $CK = \mathrm{kdf}(\sigma)$:

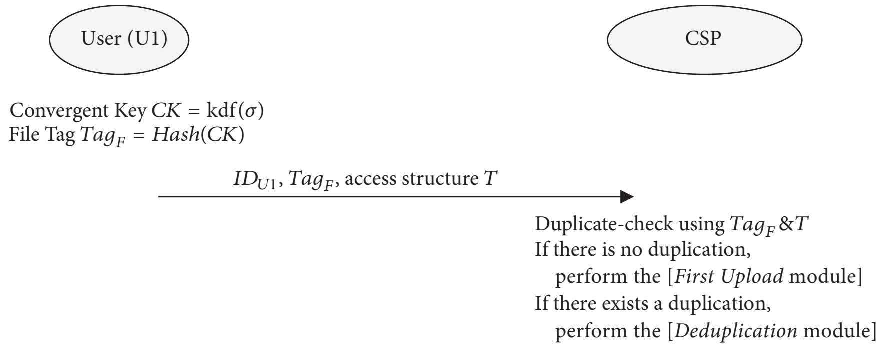

User (U1)
CSP
Convergent Key CK = kdf(σ)
File Tag $Tag_F = Hash(CK)$
ID_{U1}, Tag_F, access structure T
Duplicate-check using Tag_F &T
If there is no duplication,
perform the [*First Upload* module]
If there exists a duplication,
perform the [*Deduplication* module]

Figure 4: File upload process.

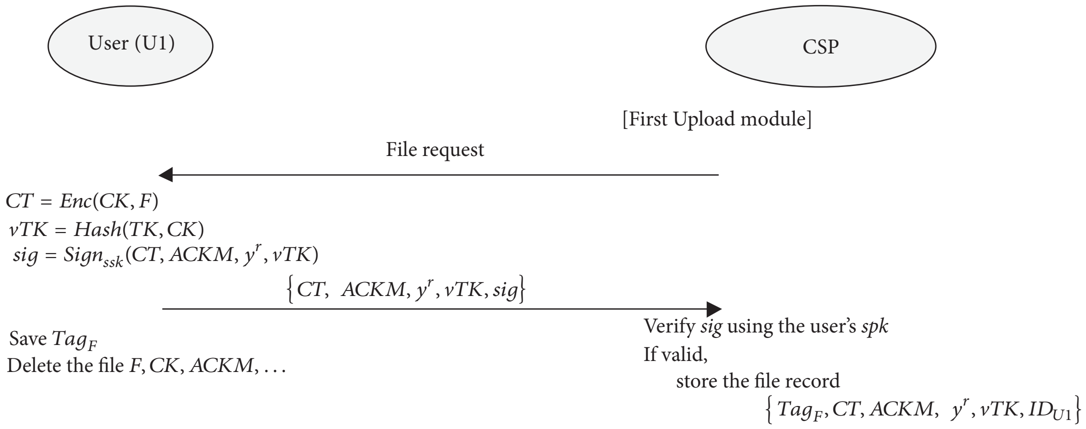

User (U1)
CSP
[First Upload module]
File request
$CT = Enc(CK, F)$
$vTK = Hash(TK, CK)$
$sig = Sign_{ssk}(CT, ACKM, y^r, vTK)$
$\{CT,\ ACKM, y^r, vTK, sig\}$
Save Tag_F
Delete the file F, CK, $ACKM$, . . .
Verify *sig* using the user's *spk*
If valid,
store the file record
$\{Tag_F, CT, ACKM,\ y^r, vTK, ID_{U1}\}$

Figure 5: First upload module.

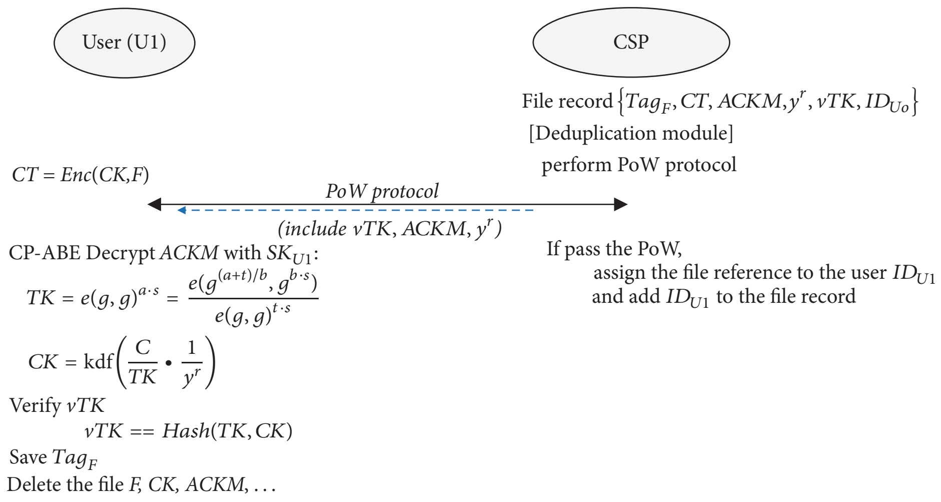

User (U1)
CSP
File record $\{Tag_F, CT, ACKM, y^r, vTK, ID_{Uo}\}$
[Deduplication module]
perform PoW protocol
$CT = Enc(CK,F)$
PoW protocol
(include vTK, ACKM, y^r)
CP-ABE Decrypt *ACKM* with SK_{U1}:
$TK = e(g,g)^{a \cdot s} = \dfrac{e(g^{(a+t)/b}, g^{b \cdot s})}{e(g,g)^{t \cdot s}}$
$CK = \mathrm{kdf}\left(\dfrac{C}{TK} \bullet \dfrac{1}{y^r}\right)$
Verify *vTK*
$vTK == Hash(TK, CK)$
Save Tag_F
Delete the file *F, CK, ACKM*, . . .
If pass the PoW,
assign the file reference to the user ID_{U1}
and add ID_{U1} to the file record

Figure 6: Deduplication module.

(a) compute $TK = e(g,g)^{a \bullet s} = e(g^{(a+t)/b}, g^{b \bullet s})/e(g,g)^{t \bullet s}$
(b) compute $\sigma' = C/TK$
(c) compute $\sigma = \sigma' \bullet y^{-r} = h^x$

(v) $U1$ verifies vTK. If pass, $U1$ obtains the file $F = Dec(CK, CT)$. Finally, verify the file as follows: $e(Hash(F), y) = e(\sigma, g)$.

4. Optimization and Discussions

4.1. Optimization. One problem to be considered is that, in the process of uploading duplicate files, the user with the duplicated file must be able to ensure that the convergent key can be correctly restored through the *ACKM* which is derived by the first uploader and is stored in the *CSP*. One possible solution is to verify the validity of the *ACKM* in the

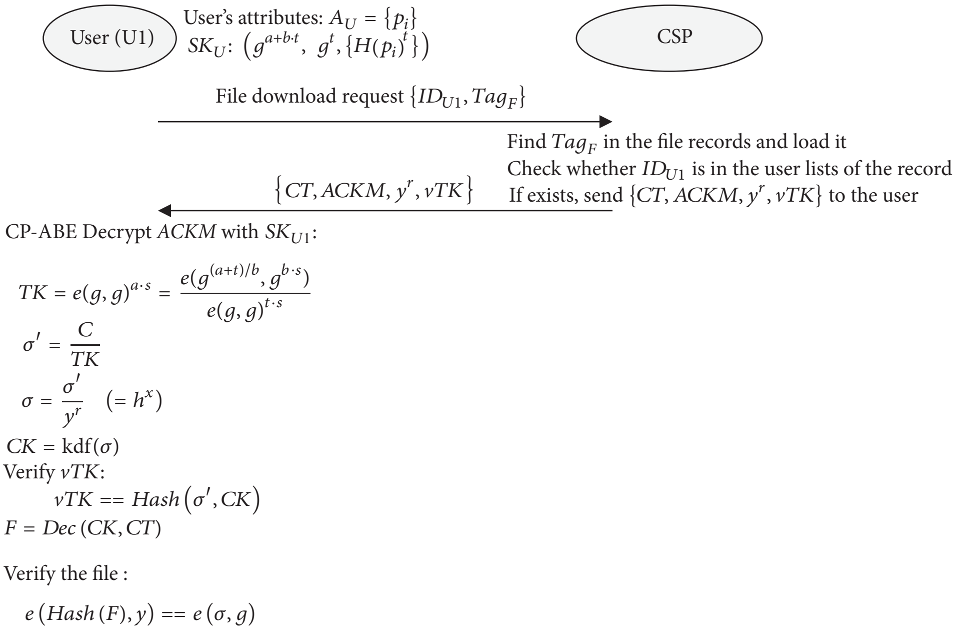

User (U1)
User's attributes: $A_U = \{p_i\}$
SK_U: $\left(g^{a+b\cdot t},\ g^t, \{H(p_i)^t\}\right)$
CSP
File download request $\{ID_{U1}, Tag_F\}$
Find Tag_F in the file records and load it
Check whether ID_{U1} is in the user lists of the record
If exists, send $\{CT, ACKM, y^r, vTK\}$ to the user
$\{CT, ACKM, y^r, vTK\}$
CP-ABE Decrypt $ACKM$ with SK_{U1}:
$TK = e(g,g)^{a\cdot s} = \dfrac{e(g^{(a+t)/b}, g^{b\cdot s})}{e(g,g)^{t\cdot s}}$
$\sigma' = \dfrac{C}{TK}$
$\sigma = \dfrac{\sigma'}{y^r} \quad (= h^x)$
$CK = \mathrm{kdf}(\sigma)$
Verify vTK:
$vTK == Hash\left(\sigma', CK\right)$
$F = Dec\,(CK, CT)$
Verify the file :
$e\left(Hash\,(F), y\right) == e\,(\sigma, g)$

Figure 7: File download process.

PoW protocol because the PoW protocol must be performed in the deduplication process. We do not describe the specific PoW protocol in this paper, but as a simple solution to this purpose, we introduced the vTK value into the PoW protocol. The verification of the validity of the $ACKM$ can be performed using the value of vTK in the PoW process. To do this, the vTK value must be stored together as an element of the $ACKM$ in the first upload process. When the CSP delivers the vTK value to the user in the deduplication process, the user can verify that the value of $\sigma' = C/TK$ $(= C/e(g,g)^{a \bullet s})$ derived from his own privilege information is valid. Then, it can be confirmed that the convergent key can be derived through the $ACKM$ stored in the CSP. In order to support this, one hash computation overhead is added in the first upload process. In the CSP, there is additional storage overhead of vTK value. Finally, in the deduplication process, one hash computation overhead for the verification of vTK value and the cost for evaluating the convergent key are added.

Also, another consideration is the possibility of optimizing metadata for the duplicated file. If the file F with access structure T is already stored in the CSP with Tag_F, a user with a different access structure T' may request the same file F to be uploaded with Tag'_F $(== Tag_F)$. That is, if different access attributes are specified for the same file, there should be some consideration as to whether to deduplicate it. The file upload process considering this can be performed as follows: for the file upload request of the user, the CSP checks whether the corresponding file tag Tag'_F exists and performs "First upload module" if it does not exist (that is, if there is no duplication). If the corresponding file tag exists (in case of duplicate upload), the access structure T' is compared. It works differently depending on the inclusion relation between the existing access structure T and the newly requested access structure T'.

(i) In case of $T' \subseteq T$: In this case, since the access structure T' of the file to which the duplicate upload is requested is included in the access structure T of the already stored file, it is not necessary to upload it because it can be restored by the information of the file record already stored in the CSP. That is, the PoW protocol is executed within the deduplication module to confirm the retention of the file, and then the access rights to the file record stored in cloud are assigned without uploading the file.

(ii) In case of $T \subseteq T'$: In this case, since the access structure T of the already stored file is included in the access structure T' of the file requested to be uploaded, it is necessary to update the existing file record in order to save only one file in the CSP. To do this, the user uploads related data such as the ciphertext CT' computed with the access structure T'. The CSP replaces the information of the existing file record with the uploaded information. Existing users can restore the file through the updated file record.

(iii) If not for the above two cases: This is the case where the access structure T and T' do not satisfy the inclusion relation. In this case, the user uploads $\{ACKM', y^{r'}\}$, and the CSP adds this information to the existing file record corresponding to the file tag Tag_F. In this case, when the user requests to download the file, the CSP sends the appropriate $\{ACKM', y^{r'}\}$ corresponding to the requesting user. This allows the user to restore the file.

4.2. Security Analysis. Our scheme provides an authorized deduplication of encrypted data stored in cloud. The security of the proposed scheme depends on the security of the used cryptographic primitives such as hash functions, symmetric and asymmetric encryption algorithms. In the security of the proposed scheme, the *AS* plays an important role, and so it is assumed that the *AS* does not collude with an attacker.

(1) The Proposed Scheme Provides an Authorized Deduplication

Proof. The proposed scheme deduplicates the files with the same contents and the proper privileges. The same files always derive the same value σ ($= h^x$, where $h = H(F)$). However, σ is distributed to the user in encrypted form using CP-ABE with the access structure T by the *AS*. The file tag Tag_F for verifying duplication is derived from the convergent key CK which is computed from σ. So, when the users have the proper privilege the same convergent key CK is derived and therefore deduplication is possible. An unauthorized user cannot obtain the same convergent key CK because of the property of CP-ABE. To get CK, an attacker needs to get σ. To do this, it has to calculate CP-ABE decryption or know the *AS*'s private key x. If an attacker does not have the proper access privileges or know the *AS*'s private key x, he cannot compute CP-ABE $Dec()$ and obtain σ. Thus, our scheme provides an authorized deduplication under the assumption of the security of CP-ABE scheme and the private key.

When a new user $U1$ with privileges equal or higher than an old user $U0$ uploads the same file that is already stored in cloud, the CSP performs the deduplication process. The security of the proposed scheme against offline brute-force attack is guaranteed by the convergent key generation procedure as in [5]. In order for an attacker with lower permissions than an old user $U0$ to perform online brute-force attacks, it is necessary to forge the tag on the file. The unforgeability of the file tag is guaranteed by (2) below. Thus, the proposed scheme provides secure deduplication. □

(2) The Proposed Scheme Guarantees the Unforgeability of the File Tag

Proof. In the proposed scheme, the user generates the file tag $Tag_F = Hash(\mathrm{kdf}(\sigma))$ through the blind signature scheme with the *AS*. The user sends m $(= h \bullet g^r)$ to the *AS*, and the AS blindly signs it, $\sigma' = m^x$ with its own private key x. And then the *AS* encrypts by CP-ABE $Enc()$ with the access structure T. An unauthorized user has to obtain a valid σ in order to forge the file tag. This requires obtaining the *AS*'s private key or breaking the CP-ABE. Therefore, if the private key of the *AS* is kept secure and the CP-ABE algorithm is secure, the unforgeability of the file tag is guaranteed. □

(3) The Proposed Scheme Guarantees That Only Eligible Users Can Access the Plain Data. That Is, It Provides the Confidentiality of Data Stored in Cloud

Proof. An attacker who colludes with the *CSP* can access an encrypted data CT and metadata Tag_F. However, the proposed method can ensure security of a plaintext data even with a low entropy because CT has encrypted with a key CK derived by interacting with the *AS* through the authorized convergent key generation process. The convergent key CK is derived by a value to which the private key of the *AS* is applied, which is encrypted and distributed by CP-ABE algorithm with the user's privilege attributes. As long as an attacker does not know the AS's private key or cannot break CP-ABE algorithm, he cannot obtain CK and therefore the confidentiality of data is preserved if the encryption algorithm $Enc()$ is secure.

A legitimate user behaving as a malicious attacker can threaten the *AS* and the *CSP*. When attacking the *AS*, an attacker has to address the discrete logarithm problem to obtain the private information of the *AS*, and so it can ensure the security of the *AS*. When attacking an encrypted data stored in the CSP, an attacker has to first pass a proofs of ownership protocol for the file. To do this in the proposed scheme, an attacker has to construct a proof for the encrypted file in the PoW process. Thus it depends on the security of the PoW protocol used (for example the Merkle Tree-based proofs of ownership scheme presented in [13] can be used). That is, for an attacker who does not own the file, it provides the confidentiality of data stored in cloud. □

4.3. Discussions. Li et al. [8] first proposed the deduplication scheme applying the privilege information. In this scheme, the file authentication tags for the deduplication are computed by applying the privilege information and these tags are generated by the *AS*. The *AS* possesses the privilege private key corresponding to the user's privileges and this privilege private key does not be distributed to the user.

While Li et al. scheme supports only simple access attributes, the proposed scheme can support hierarchical access attributes, and so direct comparisons are difficult. Li et al. scheme [8] has the following shortcomings. First, each access privilege is represented by a private key. If a user has multiple access privileges, the *AS* needs to keep private keys securely as many as the number of access attributes, which may cause a great deal of trouble in the user key management in the *AS*. Also, when a user uploads a file F that has assigned n access privileges, the *AS* needs to generate n file tags for F and sends them to the user. Then, the user must send these tags to the *CSP* for the file uploading, so it causes large network traffic. In addition, Li et al. scheme has a disadvantage in terms of the number of keys to be stored in the cloud server when compared with the proposed scheme. For a duplicated file, Li et al. scheme has to store keys as many as the number of users. In the case of Li et al. scheme, each user U stores the encrypted convergent key eK in the cloud by encrypting the convergent key CK (i.e., $eK = Enc(sk_U, CK)$) using his secret key sk_U, so that the number of keys stored in the cloud is equal to the number of users. However, the proposed scheme only needs to store one encrypted key material, $ACKM$. The authorized user can restore the convergent key via $ACKM$. Meanwhile, Li et al. scheme has another disadvantage that the *AS* is always involved in the file upload process. This adds a lot of overhead to the *AS*. In the proposed scheme, the *AS* participates only in the convergent

Table 1: Comparison of the storage overhead on the user and the *CSP*.

	Li et al.' scheme [8]	Proposed scheme
User	$N_u * (2 * N_{ap} * \text{h\_len})$	$1 * (ep\_len * (N_{ap} + 2)) + N_u * (1 * \text{h\_len})$
CSP	$N_{csp} * \{N_{ap} * \text{h\_len} + 1 * \text{ct\_len} + N_d * (N_{ap} * \text{senc\_len})\}$	$N_{csp} * \{1 * \text{h\_len} + 1 * \text{ct\_len} + 1 * (ep\_len * (N_{ap} + 2)) + 1 * ep\_len + 1 * \text{h\_len}\}$

Notes. N_u: the number of files owned and uploaded by one user; N_{csp}: the number of files that are uploaded by all users and managed on CSP; N_d: the number of deduplicated file uploads; N_{ap}: the number of access privileges assigned to the file by the user; ct_len: the bit-length of the encrypted file; ep_len: the bit-length of the elliptic curve pairings; h_len: the bit-length of the output of the hash function; senc_len: the bit-length of the output of the symmetric encryption.

Table 2: The bit-length overhead on the user and the *CSP*.

	Li et al.' scheme [8]	Proposed scheme
User	512,000 bits	28,672 bits
CSP	115,360,000 bits (110.02 Mbits)	103,840,000 bits (99.03 Mbits)

key generation process and thereafter does not involve in the protocol. This reduces the processing burden of the *AS*.

Of course, the proposed scheme that supports more complex and hierarchical privileges has higher computational complexity than existing schemes. The application of CP-ABE scheme to the convergent key generation process has the advantages of higher security and the utilization of hierarchical privileges, but the computational overhead has increased. This computational overhead does not pose a problem for the practicality of the proposed scheme. According to [17], it is reported that the encryption and decryption take about 3.3 *ms* and 6.2 *ms*, respectively, in the case of the lightweight CP-ABE scheme (the number of user attributes is 600). Thus, in case of applying the lightweight CP-ABE scheme such as [18, 19], this amount of computing costs will not be a big problem for practical use.

We compare the storage complexity of the proposed scheme with the existing scheme. Table 1 shows the storage overhead on the user and the *CSP*.

In order to make the comparison result more clear, suppose we have chosen the following bit-length as key lengths of the cryptographic primitives to ensure 128-bit security (see [20]): 128-bit for the symmetric encryption algorithm, 256-bit for the hash function and 256-bit for the elliptic curve pairings. And let N_u=100, N_{csp}=1000, N_d=10, N_{ap}=10, and *ct_len*=100,000. Table 2 shows the specific storage overhead of both schemes.

On the user side, it can be seen that the storage complexity of the proposed scheme is reduced by 483,328 bits (That is, it can save about 94%). On the CSP, the proposed scheme is about 11,520,000 bits small, so it can save about 10%. These results show that the proposed scheme has advantages in terms of storage space complexity.

The advantages of the proposed scheme are further clarified by the results of the comparison below. Figure 8 shows the storage overhead on the CSP when only the N_d (the number of deduplicated file uploads) value is increased in the above comparison. Figure 9 also shows the storage overhead when the N_{ap} (the number of access privileges assigned to the file by the user) value increases. If the number of duplicate files or the number of access privileges increases, we can see that the proposed scheme is more advantageous than the scheme of [8].

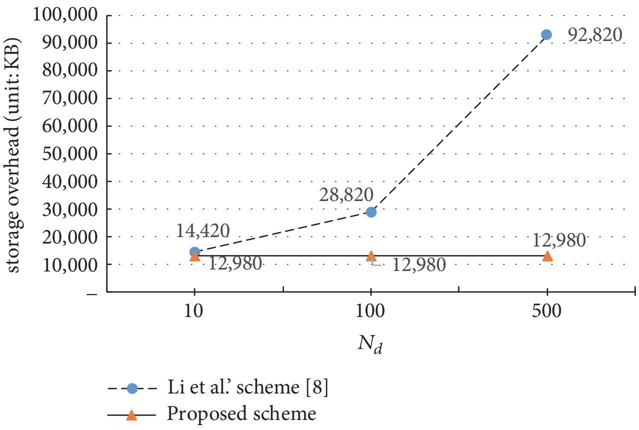
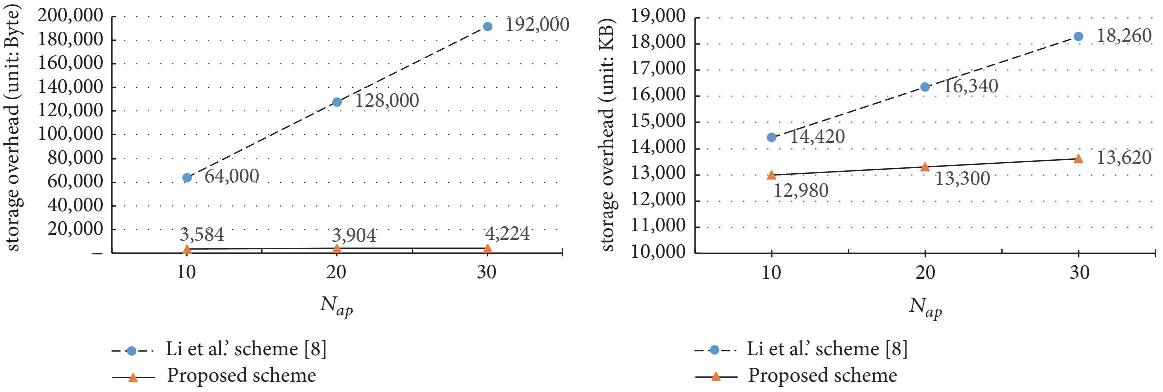


Figure 8: Storage overhead on the CSP as the N_d value changes.

The proposed CP-ABE-based authorized convergent encryption scheme can apply much more complex and various attribute privileges than existing schemes. Because it can utilize a hierarchical access structure that can be used in the CP-ABE techniques, it is very suitable for data sharing services in cloud storage. Since the convergent encryption key generated by the proposed scheme can be derived only by the authorized user, the leakage of the file information by the unauthorized user in the deduplication process can be blocked.

5. Conclusion

In this paper, we proposed the authorized deduplication scheme using CP-ABE. The proposed scheme provides client-side deduplication while providing confidentiality through client-side encryption to prevent exposure of users' sensitive data on untrusted cloud servers. Also, unlike existing convergent encryption schemes, it provides authorized convergent encryption by using CP-ABE to allow only authorized users to access critical data. The proposed CP-ABE-based authorized convergent encryption scheme can apply much more complex and various types of attribute privileges than existing schemes. Also, it satisfies security requirements. And the proposed scheme has advantages over existing scheme in



(a) On the user

(b) On the CSP

Figure 9: Storage overhead as the N_{ap} value changes.

terms of the AS's burden and storage overhead. The proposed scheme is very suitable for data sharing services in the enterprise's hybrid cloud storage model.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [18ZH1200, Core Technology Research on Trust Data Connectome].

References

[1] H. Cui, R. H. Deng, Y. Li, and G. Wu, "Attribute-based storage supporting secure deduplication of encrypted data in cloud," *IEEE Transactions on Big Data*, 2017, Early Access.

[2] P. Mell, J. Shook, R. Harang, and S. Gavrila, "Linear time algorithms to restrict insider access using multi-policy access control systems," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 8, no. 1, pp. 4–25, 2017.

[3] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Proceedings of the 22nd International Conference on Distributed Systems (ICDCS 2002)*, pp. 617–624, IEEE, Vienna, Austria, 2002.

[4] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 40–47, 2010.

[5] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage," in *Proceedings of the 22nd USENIX Security Symposium (USENIX Security 13)*, vol. 12, pp. 179–194, USENIX, Washington, DC, USA, 2013.

[6] Y. Duan, "Distributed key generation for encrypted deduplication: Achieving the strongest privacy," in *Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security (CCSW'14)*, pp. 57–68, ACM, Scottsdale, Arizona, USA, 2014.

[7] M. Miao, J. Wang, H. Li, and X. Chen, "Secure multi-server-aided data deduplication in cloud computing," *Pervasive and Mobile Computing*, vol. 24, pp. 129–137, 2015.

[8] J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1206–1216, 2015.

[9] T.-Y. Youn, K.-Y. Chang, K. H. Rhee, and S. U. Shin, "Authorized client-side deduplication using access policy-based convergent encryption," *Journal of Internet Technology*, vol. 19, no. 4, pp. 1229–1240, 2018.

[10] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Advances in Cryptology – EUROCRYPT 2005*, vol. 3494, pp. 457–473, Springer-Verlag, LNCS, Aarhus, Denmark, 2005.

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

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, IEEE, Berkeley, CA, USA, 2007.

[13] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS'11*, pp. 491–500, ACM Press, 2011.

[14] T. Youn, K. Chang, K. Rhee, and S. U. Shin, "Efficient client-side deduplication of encrypted data with public auditing in cloud storage," *IEEE Access*, vol. 6, pp. 26578–26587, 2018.

[15] C. F. Kerry, A. Secretary, and C. R. Director, "FIPS PUB 186-4 federal information processing standards publication digital signature standard (DSS)," *National Institute of Standards and Technology (NIST)*, 2013.

[16] M. Dworkin, *Recommendation for Block Cipher Modes of Operation Methods and Techniques*, NIST, USA, 2001, NIST-SP-800-38A.

[17] V. Odelu, A. K. Das, M. Khurram Khan, K.-K. R. Choo, and M. Jo, "Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts," *IEEE Access*, vol. 5, pp. 3273–3283, 2017.

[18] Y. Zhang, D. Zheng, and X. Chen, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Provable Security*, vol. 8782, pp. 259–273, Springer, 2014.

[19] H. Tsuchida, T. Nishide, and E. Okamoto, "Expressive ciphertext-policy attribute-based encryption with fast decryption," *Journal of Internet Services and Information Security (JISIS*, vol. 8, no. 4, pp. 37–56, 2018.

[20] E. Barker and A. Roginsky, "Transitioning the use of cryptographic algorithms and key lengths," *Draft NIST Special Publication 800-131A Revision 2, National Institute of Standards and Technology (NIST)*, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 6974809, 11 pages
https://doi.org/10.1155/2019/6974809

Research Article

A Multiuser Identification Algorithm Based on Internet of Things

Kaikai Deng, Ling Xing, Mingchuan Zhang, Honghai Wu, and Ping Xie

School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Correspondence should be addressed to Ling Xing; xingling_my@163.com

Received 18 January 2019; Revised 2 April 2019; Accepted 11 April 2019; Published 2 May 2019

Guest Editor: Vishal Sharma

With the rapid development of the Internet of Things (IoT) in 4G/5G deployments, the massive amount of network data generated by users has exploded, which has not only brought a revolution to human's living, but also caused some malicious actors to utilize these data to attack the privacy of ordinary users. Therefore, it is crucial to identify the entity users behind multiple virtual accounts. Due to the low precision of user identification in the many-to-many mechanism of user identification, a random forest confirmation algorithm based on stable marriage matching (RFCA-SMM) is proposed in this study. It consists of three key steps: we first employ the stable marriage matching model to calculate the similarity between multiple users and utilize a scoring model to calculate the overall similarity of the users, after which candidate matching pairs are selected; second, we construct the random forest model that exploits a user similarity vector training set; afterward, the candidate matching pairs combine the secondary confirmation of the random forest model, which both improve the precision of the many-to-many user identification and protect private user data in the IoT. Extensive experiments are provided to demonstrate that the proposed algorithm improves precision rate, recall rate, and F-Measure (F1), as well as Area Under Curve (AUC).

1. Introduction

As 4G/5G technology continues to evolve, it provides people with more efficient performance and increased speed in order to meet higher standards of data services for more users. At the same time, the higher speed and more reliable transmission of mobile communication further promote the development of the Internet of Things (IoT) in the large-scale era. The amount of user data is constantly increasing in the IoT context. Leveraging user data to analyze the social behavior of users can enable the provision of a safer social environment. According to statistics, 42% of users use multiple social networks simultaneously [1]. The IoT integrates different social methods to meet people's different needs to the greatest extent possible [2]. For instance, RenRen and Sina Microblog are services used to share personal statuses and publish blogs anytime and anywhere in China. However, as there is no direct link between user data on these services, a complete social network map is difficult to obtain. Multiuser identification is therefore employed to identify users of multiple virtual accounts [3, 4], allowing user data to be better protected in the IoT era.

User identification is also referred to as user matching. Many studies have addressed the user identification problem by examining user profile information attributes, primarily the user's personal information and published content, which includes username, geographical location, blog posts, etc. [5–13]. Although missing data is an issue in the process of filling in these attributes, they can still be filled in through the use of appropriate methods. Moreover, some attributes play an extremely important role in the process of user identification. Therefore, user identification based on user attribute information can better accomplish the work of identification. Some of the research on this topic focuses on the use of network topology for user identification. This research mainly relies on the user's circle of friends to identify a specific user [14–19]. The similarity between user accounts is judged by analyzing nodes between users. However, due to the heterogeneity of the network structure in practical applications, this method requires improvement in terms of its precision.

This study is divided into three sections: user profile data, user-generated data, and user-associated data, according to the type of IoT data involved. User profile data refers to the data that the user needs to enter or select when they register for their accounts. User-generated data refers to the data that is the user's published content. User-associated data refers to the data that users associate with other users. Among the numerous user attributes, some attributes are more important to the task of user identification, such as web links, blog posts, etc. Conversely, some attributes such as gender, age, etc. are less important in this context. Therefore, reasonable allocation of corresponding weights to users can also improve the precision of user identification to a certain extent.

A stable marriage matching algorithm is mainly used to find and solve a stable matching pair problem and has been widely used in the fields of economics and computing for this purpose [20]. Stable matching refers to cases in which the two identified items are mutually optimal choices rather than the first choice in the current match and where there is no better match to any one element in the unmatched elements. Meng Bo proposed that the ranking-based cross-matching algorithm could also adopt the concept used in the stable marriage matching algorithm. The algorithm uses profile attribute similarity (PAS) and user surrounding score (USS) to select candidate matching users and then uses the user matching score (UMS) to determine the users that match with the candidate users. In order to further improve the matching precision, a cross-matching process inspired by the stable marriage matching algorithm is added. Finally, the matching user pairs are obtained by a simple pruning process. However, the idea behind the stable marriage matching algorithm is that all users should match; thus, it is difficult to guarantee the precision of the final result. The existing user identification algorithm, which is based on supervised learning, is guaranteed in accuracy, but can only achieve one-to-one user matching.

Accordingly, a random forest confirmation algorithm based on stable marriage matching (RFCA-SMM) is proposed in this study. The proposed algorithm combines stable marriage matching and a scoring model to obtain the overall similarity of users in addition to candidate matching pairs. The similarity calculation of user attribute data is used to construct the user similarity vector, while the training set of the user similarity vector is leveraged to generate the random forest model. The final matching results of candidate matching pairs can be obtained by means of random forest confirmation.

2. Related Works

Multiuser identification technology is significant in both research and practice in many important fields. Current studies of multiuser identification can be divided into three categories according to the way feature information is used: user identification based on user profile information, network structure, and user-generated information.

2.1. User Profile Information-Based User Identification. Research based on user profile information for the purpose of solving user identification problems primarily focuses on personal information. The classification model is constructed, after which the corresponding matching strategy is used to complete user identification. Raad et al. [20] proposed the Friend of a Friend (FOAF) attribute matching strategy. Ye et al. [21] proposed an objective weighting method for user attributes to integrate user attribute information and complete the calculation of user profile similarity. Cortis et al. [8] proposed an identity recognition algorithm that assigns weights to individual attributes of user profile information and then calculates the similarity among attributes with reference to both grammatical and semantic aspects. Able et al. [22] aggregated user profile information in order to match users. Zamani et al. [23] took the user's unit, interests, and other attribute information into full consideration, integrating the similarity of multiuser attributes via the equal evaluation model and complex mixed training model; this improves the possibility of correctly identifying users owing to the personalized characteristics of many users' attribute information. Therefore, leveraging user attribute information to achieve user identification is a good choice.

2.2. Network Structure-Based User Identification. Network structure-based studies on user identification mainly focus on recognizing identical users by examining the user's circle of friends. The user's friend relationships are easy to obtain, the problem of malicious imitation and forgery is less likely to occur, and the importance of information coupling of local topology on network development has been certified. Narayanan et al. [14] proved for the first time that user identification can be accomplished by relying on the topology structure of the network; however, the precision of the matching results required improvement. Bartunov et al. [15] proposed the construction of the objective function by combining attribute information and network structure information and then optimized the function to obtain the optimal matching pair. Cui et al. [16] integrated user profile information similarity and graph similarity to achieve mapping from an email network to a Facebook network. Liu et al. [17] proposed the HYDRA approach to modeling user behavior by employing user attributes and user-generated content. Korula et al. [18] abstracted the problem of user identification into a mathematical definition, arguing that different social networks are generated by user graph structure through probability and that the selection process of graph edges is one of approximate probabilities. Tan et al. [19] modeled users' social relations and mapped users to low-dimensional space to improve the efficiency of user identification in the network. However, there is heterogeneity between nodes in the actual network structure, and the influence of this heterogeneity is ignored in the calculation process; therefore, the precision of this method in the context of user identification requires improvement.

2.3. User-Generated Information-Based User Identification. User identification based on user-generated information mainly relies on the content published by users. Now that the Internet of Things (IoT) has a close relationship with our daily lives, people can immediately post their own dynamic content and comment on the content posted by the friends around them. As user behavior habits are not easy to change and hide [24], these habits can correspond strongly with the characteristics of the users themselves. Therefore, the use of data mining algorithms to discover these hidden association rules [25] can greatly improve the recognition rate of user identification. Goga et al. [10] used the geographic locations, timestamps, and writing habits of users' published statuses for user identification purposes. Li et al. [13] proposed a supervised machine learning algorithm to solve the user identity recognition problem based on user-generated content. In recent years, the development of mobile communication technology has made a great contribution to the incorporation of geo-tagging when users publish their statuses. As the user's track of action is not easy to imitate, the application of geographical location attributes to user identification opens up a new method of identifying users. Cao et al. [26] proposed an identification method for processing multisource data by utilizing the cooccurrence frequency of two user trajectories. Hao et al. [27] proposed that user trajectories are transformed into sequences composed of multiple grids, which are in turn transformed into vectors by using a TD-IDF model, after which the similarity of user trajectories is calculated via cosine similarity. Han et al. [28] proposed that each geographic coordinate point should be represented as a corresponding semantic position. The user's trajectory can thus be represented by the text composed of the semantic position, with the LDA model being used to represent the user's topic distribution; finally, the similarity of the user trajectories is calculated to determine whether the two users are the same. Therefore, the analysis of user behavior information for multiuser identification is ideal in this context.

3. Data Preprocessing

3.1. Filling Missing User Data. Data filling is commonly applied to user profile data processing. When a user registers an account, data may be missing for various reasons such as, e.g., privacy protection. Therefore, an appropriate filling method should be adopted for each different type of data from each dimension, as follows:

(1) Similarity filling: filling in user data by utilizing the relational degree [29] between other users and users with missing data. For example, user A and user B are friends, and they will generate social behaviors such as comments, reposts, and thumb up on social networks, where the comments indicate that the content posted by the friend on the social network is explained, reposts indicate that friends have similar interests, and thumb up indicates that they agree with the content posted by friends. The relational degree between users is calculated through the behaviors of "comment C_{AB}", "repost R_{AB}", and "thumb up L_{AB}" between users. The three types of user behavioral information are sequentially assigned the weights "3", "2", and "1". Select n users with the highest relational degree and take the mode number for filling. If the user with missing data has a low relational degree with other users, the data will not be filled. The relevant formula is as follows:

$$R_{AB} = 3\sum C_{AB} + 2\sum R_{AB} + \sum L_{AB} \tag{1}$$

(2) Speculated filling: the missing data is inferred from other attributes. This method is mainly used for user gender filling. The blog posts published by the user best reflect the characteristics of the user's personality; thus, by using the user's blog post information, the Bayesian classification model [30] can be employed to accomplish user gender speculation.

The Bayesian classification model is constructed as follows:

$$p(m \mid w_i) = \frac{p(w_i \mid m)\, p(m)}{p(w_i)} \tag{2}$$

where $p(m \mid w_i)$ denotes the probability that the user is identified as male when the word w_i occurs, $p(w_i \mid m)$ denotes the probability of the word w_i occurring in all males, $p(w_i)$ denotes the probability of the user being male, and $p(m)$ denotes the probability of the occurrence of word w_i.

Given the complexity of calculating $p(w_i \mid m)$, this article calculates the conditional independence naïve hypothesis. The formula is as follows:

$$\begin{aligned} p(w_i \mid m) &= \prod_{i=1}^{n} p(w_i \mid m) \\ &= p(w_1 \mid m)\, p(w_2 \mid m) \cdots p(w_n \mid m) \end{aligned} \tag{3}$$

$$p(w_i) = \prod_{i=1}^{n} p(w_i) = p(w_1)\, p(w_2) \cdots p(w_n) \tag{4}$$

Therefore, the Bayesian classification model constructed is as follows:

$$\begin{aligned} &p(m \mid w_i) \\ &\quad = \frac{p(w_1 \mid m)\, p(w_2 \mid m) \cdots p(w_n \mid m)\, p(m)}{p(w_1)\, p(w_2) \cdots p(w_n)} \end{aligned} \tag{5}$$

The statistical results of the training set can be used to derive the probability required for the calculation in the Bayesian classification model. The prediction results of the corresponding attributes can be obtained via this model.

3.2. User Data Similarity Calculation. In view of the problem that the user data in the IoT has a different format, the user data format needs to be generalized before the similarity between each attribute in this study can be calculated; this processed data is more suitable for similarity calculation. The relevant calculation methods are as follows:

(1) Dice coefficient [31]: when calculating strings, they can be divided into two categories. When calculating the multivalued strings n_i and n_j, the sum of the two times of the intersection information and divided by the sum of

the elements of n_i and n_j yields the two strings of Dice coefficients, which are calculated as follows:

$$Simfunc\left(n_i, n_j\right) = 2\frac{\left|n_i \cap n_j\right|}{\left|n_i\right| + \left|n_j\right|} \tag{6}$$

Example. In two multivalued attribute strings "vivid music movie" and "movie travel", the intersection information is {"movie"}, so the similarity is 2/5=0. 4.

For single-valued attribute strings, the Dice coefficient is calculated as above, except that the intersection information is different.

Example. For the single-valued strings "johe" and "joh", the intersection information is "jo, oh", so the similarity is 4/5 = 0.8.

(2) Levenshtein distance [32]: the number of character edit steps required to calculate the equality of two strings is used as an operational cost to measure the difference between strings. The formulae for calculating the similarity of strings n_i and n_j are as follows:

$$Simfunc\left(n_i, n_j\right) = 1 - \frac{d\left(n_i, n_j\right)}{\max\left(\left|n_i\right|, \left|n_j\right|\right)} \tag{7}$$

where $d(n_i, n_j)$ denotes the distance between the strings n_i and n_j and $\max(|n_i|, |n_j|)$ denotes the maximum value of characters contained in the strings n_i and n_j.

(3) Cosine similarity [33]: this is mainly used to calculate the vector composed of user attributes. Assuming that A and B are two n-dimensional vectors, such that A is $[A_1, A_2, \ldots, A_{\mathrm{n}}]$ and B is $[B_1, B_2, \ldots, B_{\mathrm{n}}]$, then the cosine value of angle θ between A and B is the similarity value between vectors. The formula is as follows:

$$\cos\theta = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^{\,2}} \times \sqrt{\sum_{i=1}^{n} B_i^{\,2}}} \tag{8}$$

The closer the angle between two vectors is to 1, the higher the similarity between two users is. The closer the angle between the two vectors is to 0, the smaller the cosine value of the included angle is and the lower the user similarity is. By comparing the size of cosine values, it can be determined whether the two accounts are identical.

(4) Term frequency-inverse document frequency (TF-IDF) [34]: this is mainly used to measure the importance of a certain word in the document and is often used to deal with multiword attribute fields such as personal profiles. The specific steps are as follows.

Step 1. Calculate the term frequency (TF) of each word in the document;

$$TF = \frac{n}{N} \tag{9}$$

where n denotes the number of occurrences of a certain word and N denotes the total number of words in the document.

Step 2. Calculate the inverse document frequency (IDF) of each word in the document;

$$IDF = \log\left(\frac{D}{P+1}\right) \tag{10}$$

where D denotes the total number of documents in the corpus, P denotes the number of documents containing a word in the document, and 1 is added to avoid cases in which the denominator is 0.

Step 3. Calculate the TF-IDF of each word in the document;

$$TF - IDF = TF \times IDF = \frac{n}{N} \times \log\left(\frac{D}{P+1}\right) \tag{11}$$

Step 4. Select keywords in each document to construct a term frequency vector for calculating similarity.

Step 5. Calculate the similarity value by cosine similarity.

(5) User blog data similarity calculation: frequent item sets of user blog data are mined by means of frequent pattern mining to calculate user similarity. Due to the difference in the amounts of user-published content, the one-item set is also used as a calculation indicator in this study. "1" is added to avoid a high-frequency item set in the calculation of similarity. The formula is as follows:

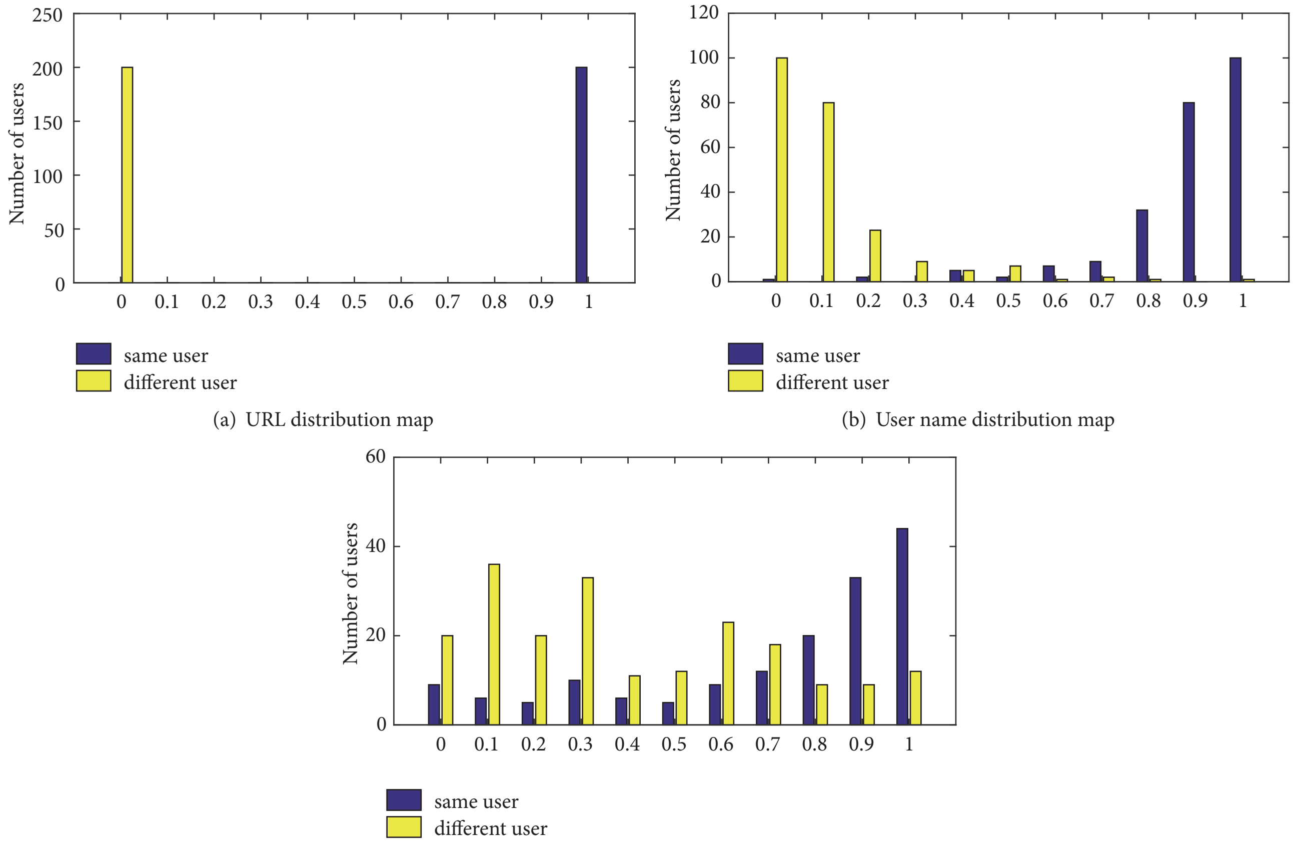
$$S_{AB} = \sum_{E_{\mathrm{i}} \in A \cap B} \left(\left(1 + CA_{E_{\mathrm{i}}}\right) \times \left(1 + CB_{E_{\mathrm{i}}}\right)\right)^{C_{E_{\mathrm{i}}}} \tag{12}$$

where CA_{E_i} denotes the support degree count of the frequent item set E_i of user A, CB_{E_i} denotes the support degree count of the frequent item set E_i of user B, and C_{E_i} denotes the item set number of E_i. The similarity threshold is set at 5,000 based on historical data. If $S_{AB} > 5000$, return "1"; otherwise, return "0".

(6) State timestamp similarity calculation: the time points of users' publishing status also have certain personalized characteristics. The average dynamic number can be obtained by dividing the dynamic number generated by users in a certain period of time by the total dynamic number. The average dynamic number is then used to form a user timestamp vector of 24 dimensions. The similarity is calculated; users are determined to be the same user when Sim<0. 1 according to the statistical results. The formula is expressed as

$$Sim_t\,(a,b) = \sum_{i=1}^{24} \left|u_{ai} - u_{bi}\right| \tag{13}$$

where u_{ai}, u_{bi} denote the average dynamic number of the ith time period of users a and b.



(a) URL distribution map

(b) User name distribution map

(c) Interests distribution map

Figure 1: User attribute similarity distribution.

4. Multiuser Identification Method

4.1. Building User Similarity Vectors. Research and analysis of user data in the Internet of Things (IoT) context can assist in meeting people's network needs. However, some malicious users will employ the user data to attack normal users via the IoT. Therefore, it is necessary to identify and analyze IoT users.

In this study, the profile information and behavior information of user data are utilized to achieve multiuser identification. After filling in user profile information, the precision of user identification can be increased to a certain extent. User behavior information has the characteristic of being personalized, which allows for highlighting of the user's own behavior habits and is thereby conducive to the improvement of user identification precision. A reasonable similarity calculation method is used for the data of each dimension of the user. The data for each dimension is provided with a threshold value when calculating similarity. After comparing the calculated similarity of attributes with the set threshold, qualified results return "1"; otherwise, return "0". Thus, user similarity vectors composed of "0" and "1" can be formed and used for the input of subsequent algorithms.

4.2. Weight Analysis of User Attributes. By calculating the similarity between user attributes, the whole similarity vector of user attributes can be obtained. As different user attributes have different influences on the degree of user recognition, it is necessary to calculate the weight of the attribute items. Figure 1 shows the performance of a single attribute in user identification. It can be clearly seen from Figure 1 that the URL and user name have different distributions of similarity between the same user and different users when user matching is performed. As these attribute items are highly distinguishable, the weight allocation should be relatively large. When users match in terms of their interests, the similarity distribution between the users is small, meaning that the weight distribution should also be small. Again, as each attribute has different effects on user identification, it is therefore necessary to assign corresponding weights to each attribute.

4.3. Weight Allocation Algorithm. After the user data is preprocessed, multiple user attributes are determined. When determining the weighting coefficient of the similarity judgment of each attribute in the user data, the traditional expert subjective weighting method encounters the problem of poor robustness, while the objective weighting method relies too much on the existence of a large amount of sample data, which is poor in universality. Therefore, this article proposes the posterior probability-based information entropy weight allocation algorithm.

```
Input: Source network account user data information vector $F_A$, user data vector $\{F_j\}_{j=1}^{k}$ for
all accounts in the target network, user data vector $F_B$ to be matched account in the target
network
Output: The final similarity $V_{final} = W_i(x)V(F_A, F_B)$ between the two accounts $F_A$ and $F_B$
foreach $F_j$ in $\{F_j\}_{j=1}^{k}$
    for i=1 to n
    Calculate the similarity $V(F_A, F_B) = (v_1^{AB}, v_2^{AB}, ..., v_n^{AB})$ of account A and B by using
formula (6) (7) (8) (11) (12) (13)
  end
  for i=1 to n
      The attribute weights of the user data are assigned using equation (15)
    end
    Calculate the final similarity $V_{final} = W_i(x)V(F_A, F_B)$ between the two accounts $F_A$ and $F_B$
    return $V_{final}$
```

Algorithm 1: User data similarity calculation.

In information theory, the entropy value reflects the degree of information disorder. The smaller its value is, the more orderly the information is, and the more valuable this attribute is; on the contrary, the more disordered the information is, the lower the value of this attribute is. Therefore, information entropy can be used to evaluate the effectiveness of the attributes used. According to the definition of information entropy, for any random variable, the formula is as follows:

$$E(x) = -\sum_{x \in X} P(x) \log P(x) \tag{14}$$

where $p(x)$ is the possible value probability for the attribute.

In order to make the probability description of attributes more precise, more effective weights are assigned to each attribute. On the basis of information entropy, the posterior probability of user attributes is further calculated, which aids in improving the precision of user identification. By combining the posterior probability and information entropy, the attribute weight of the user account is $W(x)$, such that

$$W(x) = -p(y_s \mid s) \sum_{x \in X} p(x) \log(p(x)) \tag{15}$$

where $p(y_s \mid s)$ is the posterior probability of the attribute.

The user data information contains n attribute items. The data information vector is $F_j = (a_1^j, a_2^j, ..., a_n^j)$, where a_i^j $(i = (1, 2, ..., n))$ represents the ith attribute information of account j. The user similarity vector is defined as $V(F_A, F_B) = (v_1^{AB}, v_2^{AB}, ..., v_n^{AB})$, where v_i^{AB} represents the similarity between the ith attribute of user F_A and user F_B's attribute information. If the similarity exceeds the threshold, output "1"; otherwise, output "0". Accordingly, the user similarity vector is a vector composed of "1" and "0". Therefore, the final user similarity vector is $V_{final} = W_i(x)V(F_A, F_B)$. The process is summarized in Algorithm 1.

4.4. Random Forest Confirmation Algorithm Based on Stable Marriage Matching

4.4.1. Similarity Score. In order to improve the efficiency of the similarity calculation between users, this study adopts a stable marriage matching algorithm to perform the many-to-many matching calculation. The overall similarity of users is evaluated by means of the similarity score of user matching. The relevant formula is as follows:

$$Score = \sum_{i=1}^{20} w_i v_i^{AB} \tag{16}$$

where Score denotes the final Score of the match, w_i denotes the weight of the ith attribute of the user, and v_i^{AB} denotes the similarity of the ith attribute of user A and user B. The higher the Score is, the more likely it is the same user.

4.4.2. Stable Marriage Matching Algorithm. The scoring formula can evaluate the overall similarity of two users based on user data information. The higher the score, the more likely it is that the two users are the same user. The stable marriage matching algorithm uses the similarity score between users to select candidate matching pairs. If we calculate user data for all accounts, then the computational complexity will be high. Therefore, it is necessary to obtain v_s by filtering the target account in another network according to the condition C: filter accounts by username. The specific steps involved are as follows.

Step 1. Each user on social network M and the user on social network N are matched by scoring formula.

Step 2. The user on M is matched with the user on N who ranks first according to the score. If the user on N has already matched other users, the user will compare the user who has already matched himself with the user who is requesting matching with himself. Finally, the user with the highest score will be selected as the other half of the matching pair.

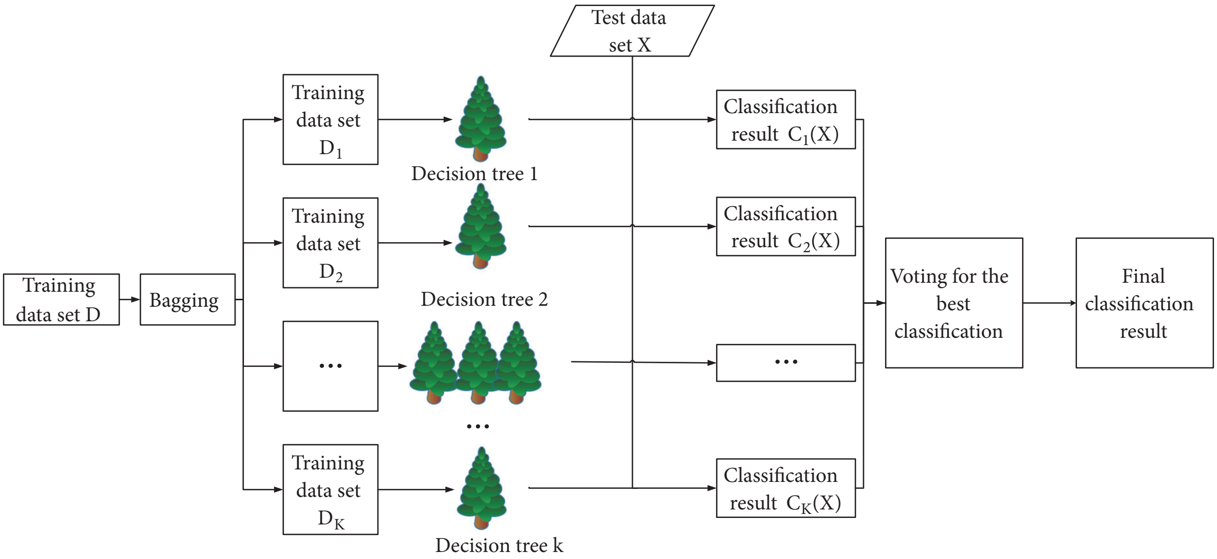


FIGURE 2: Random forest model construction process.

Step 3. After *Step 2* is complete, some users will still fail to be matched successfully. A user who does not match will be matched with the highest ranked user among all users who have not rejected themselves, after which *Step 2* will be repeated until all users match.

4.4.3. Random Forest Algorithm. Users through the stable marriage matching algorithm output is a matching pair. However, such results cannot be used directly, as it would be easy to obtain poor matches if this was the case. In order to solve this problem, a second confirmation of random forest is established in order to eliminate the negative influence of wrong matching results on the final results as far as possible.

There are many algorithms based on supervised learning, including logistic regression, SVM, Adaboost, etc. Random forest is selected as the final quadratic confirmation algorithm in this study for the following reasons.

(1) There are 20 data dimensions used in this study, among which there may be linear correlation dimensions. The dimension of linear correlation not only plays no positive role in the training of the supervised learning model, but also impacts the effect of other nonlinear correlation dimensions. In general machine learning model training, data dimensionality reduction will be processed, and data dimensionality reduction is a tedious process. However, random forests are not sensitive to multicollinearity, and the results are robust to missing and unbalanced data.

(2) While overfitting is always discussed in machine learning, it is not easy for the random forest model to produce the overfitting phenomenon owing to the randomness involved.

Figure 2 provides an overview of the construction process of the random forest validation model. The specific steps are as follows.

Step 1. Acquire the training set.

(a) The original input training data set D comprises 20 prediction attributes and a classification label Y. The 20 prediction attributes are the user similarity vector $V(F_A, F_B) = (v_1^{AB}, v_2^{AB}, ..., v_n^{AB})$ obtained in the above, while the classification label is $Y = (1 \parallel 0)$. A class label of “1” means that the users are the same, while “0” means that they are not the same.

(b) The original training data set D is sampled by random sampling with K playback times via the Bagging method, and a new training subset U of K is obtained.

Step 2. Generate the decision tree.

(a) The number of prediction attributes in the training sample is 20, and $F = \sqrt{20} \approx 5$ attributes are randomly selected from the 20 prediction attributes to form a random feature subspace X_i, which is the split attribute set of the current node of the decision tree. During the generation of the random forest model, F remains unchanged.

(b) According to the decision tree generation algorithm, each node is split by selecting the optimal split attribute from the random feature subspace X_i.

(c) Each tree grows completely without pruning. Finally, according to each training set D_i, the corresponding decision tree is generated as $h_i(D_i)$.

(d) Combine all the generated decision trees together to generate a random forest model $\{h_1(D_1), h_2(D_2), ..., h_i(D_i)\}$. Each test tree $h_i(D_i)$ is tested using the test set sample X to obtain a corresponding classification result $\{C_1(X), C_2(X), \ldots, C_K(X)\}$.

(e) Using the plurality voting method, according to the classification result output by the K-tree decision tree, the classification result with a large number of decision trees is used as the final classification result corresponding to sample X of the test set.

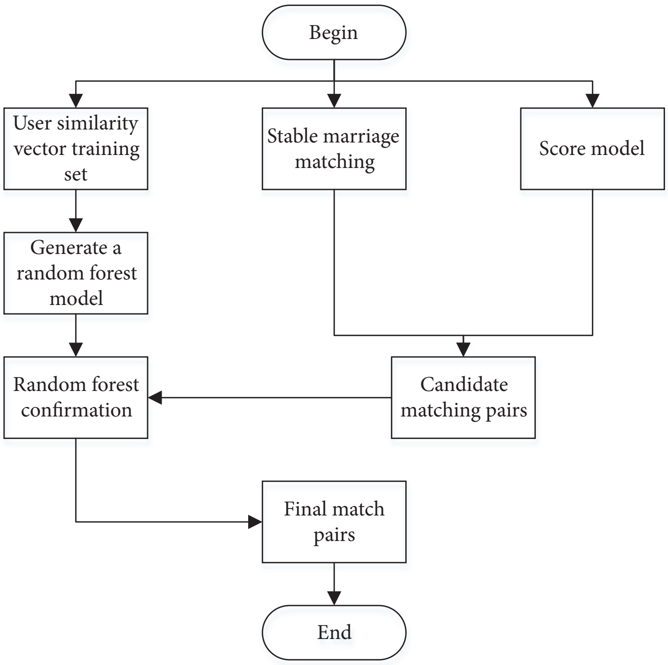

Begin
User similarity vector training set
Stable marriage matching
Score model
Generate a random forest model
Random forest confirmation
Candidate matching pairs
Final match pairs
End

Figure 3: Random forest confirmation based on stable marriage matching.

Figure 3 describes the process of RFCA-SMM. The input data of the algorithm is the user attribute data in the IoT. By using the stable marriage matching algorithm in combination with the scoring formula, the overall similarity between users can be obtained in order to select the candidate matching pairs with the user similarity vector training set as input. The random forest model is constructed, and the candidate matches obtained are used as input data to confirm and identify in the random forest. If the identification result for the candidate matching pair is not the same user, the candidate matching pair is marked as "unmatched"; by contrast, if the candidate match pair contains two instances of the same user in the random forest confirmation, the final match result is generated. The algorithm flow of RFCA-SMM is represented by Algorithm 2.

5. Analysis of Experimental Results

In order to verify the effectiveness of the proposed algorithm, [35] provides five open datasets of foreign mainstream social networks.

In this study, precision rate, recall rate, F-measure (F1), and AUC are used as evaluation criteria. The relevant formulae are as follows:

$$precision = \frac{tp}{tp + fp} \tag{17}$$

$$recall = \frac{tp}{tp + fn} \tag{18}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{19}$$

```
Input: To be matched account $v_s$, Candidate matching account $v_{match}$, the set of accounts $V_M$ that have not been matched in the social network M, the set of accounts $V_N$ that have not been matched in the social network N
Output: The final match pairs set R
R=$\phi$
Initialize unmatched queue
while $V_M \neq \phi$ and $V_N \neq \phi$ do
    if $v_s$=NULL then
        $v_s$=UserSelect($V_M$,$V_N$)
        $v_{match}$= UserMatch($v_s$,$V_M$,$V_N$)
    end if
    ($v_s$,$v_{match}$)= Secondary confirmation ($v_s$,$v_{match}$,R)
end while
pruning process
return R
```

Algorithm 2: RFCA-SMM.

AUC: the area under the Receiver Operating Characteristic (ROC) curve is directly calculated. The ROC curve is defined as the X-axis by the False Positive Rate (FPR), while the True Positive Rate (TPR) is defined as the Y-axis. The formulae for these two values are as follows:

$$TPR = \frac{tp}{tp + fn} \tag{20}$$

$$FPR = \frac{fp}{fp + tn} \tag{21}$$

where tp denotes the number of the same users that are correctly matched, tn denotes the number of users that are

Table 1: Comparison of several types of supervised learning.

Algorithms	Precision	Recall	F1	AUC
Random Forest	0. 961	0. 881	0. 919	0. 961
SVM	0. 950	0. 860	0. 903	0. 945
Logistic	0. 935	0. 900	0. 917	0. 965
Adaboost	0. 910	0. 870	0. 890	0. 900

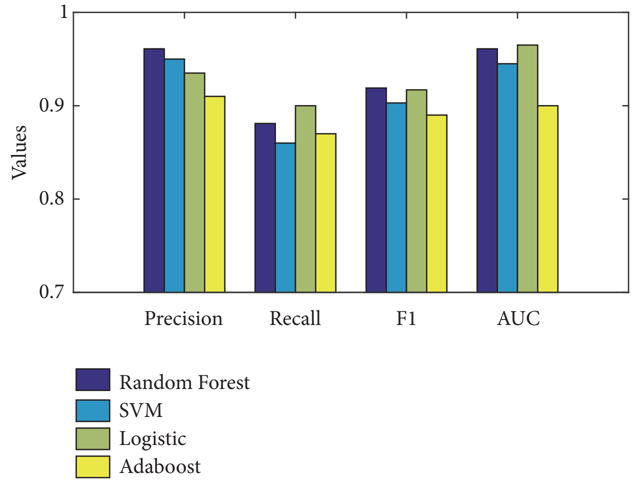


Figure 4: Comparison of several types of supervised learning.

unmatched and not the same, fp denotes the number of users that are matched but are not the same, and fn denotes the number of users that are not matched but are the same users.

5.1. Comparison of Random Forest Model and Other Supervised Learning Models. This article uses the random forest supervised learning model for the confirmation of candidate matching pairs. In order to verify the effectiveness of the proposed algorithm in obtaining the matching results, the random forest model and other supervised learning models are analyzed with reference to the evaluation indicators outlined above. The ratio of training set data to test set data is 3:1 and the number of users is 1000 pairs. The results are presented in Table 1 and Figure 4.

It can be seen from Figure 4 that these supervised learning algorithms have relatively good results; among them, the best performing algorithms are Random Forest and Logistic. Random Forest performs slightly better in precision rate and F1, while Logistic performs slightly better in recall rate and AUC. However, considering the modeling scenarios of these two supervised learning algorithms, Random Forest has an advantage over Logistic. Therefore, the effectiveness of the random forest model is also proven.

5.2. Comparison of RFCA-SMM and RCM Algorithm Results. This section presents a comparative analysis of the RFCA-SMM and Ranking-based cross-matching (RCM) algorithms. The purpose of the RCM algorithm is to accurately find more matching pairs, which decompose the seed user's identification into a step-by-step iterative process. In the iteration process of each step, the calculation process of the algorithm is divided into three substeps: account selection, account matching, and cross matching. Accumulate the results of each iteration to form a result set. The advantage of this algorithm is to compare the results obtained each time and select the user account with a high score as the final result. However, the precision of the RCM algorithm is largely affected by the number of seed users (that is, the number of users who are known to match pairs). If it is not possible to know in advance which accounts are the same user (that is, there is untagged identity match), then the RCM algorithm needs to be improved in terms of precision. Since the algorithm for user identification in this study is untagged, the experimental results of the two algorithms are analyzed using an unmarked data set, as shown in Table 2 and Figure 5.

Table 2: Comparison of RFCA-SMM and RCM results.

Algorithms	Precision	Recall	F1	AUC
RFCA-SMM	0. 962	0. 871	0. 914	0. 961
RCM	0. 934	0. 875	0. 904	0. 912

It is clear from Figure 5 that the proposed RFCA-SMM algorithm is superior to the RCM algorithm in terms of precision, F1, and AUC in this study, although its performance is slightly lower than that of the RCM algorithm in terms of recall rate. The reason is that the proposed algorithm performs user-generated data processing, user attribute weight distribution, and secondary confirmation based on supervised learning compared with the RCM algorithm. It can be seen from Figure 5 that the final user matching results of the proposed algorithm achieve some improvement in the evaluation index compared with the RCM algorithm, mainly because the proposed algorithm (unlike RCM) does not take the social network structure into account [36]. In summary, the algorithm proposed in this study improves the precision of user identification to a great extent.

6. Conclusions

The key features of 4G/5G technology, namely, low energy consumption and low delay, have laid the foundation for the development of the Internet of Things (IoT). Given the various other advantages of this technology, it can effectively promote the rapid development of the IoT industry chain. Since most of the information in the IoT is related to private user data, we propose a random forest confirmation algorithm based on stable marriage matching (RFCA-SMM). The candidate matching pairs are obtained by combining the stable marriage matching algorithm with the scoring model. In order to demonstrate the effectiveness of the random forest model, we analyze the random forest model and several other supervised learning models on the evaluation indicators and finally the second confirmation of candidate matching pairs via the random forest. Moreover, we conduct a comparative analysis of RFCA-SMM and RCM. The experimental results

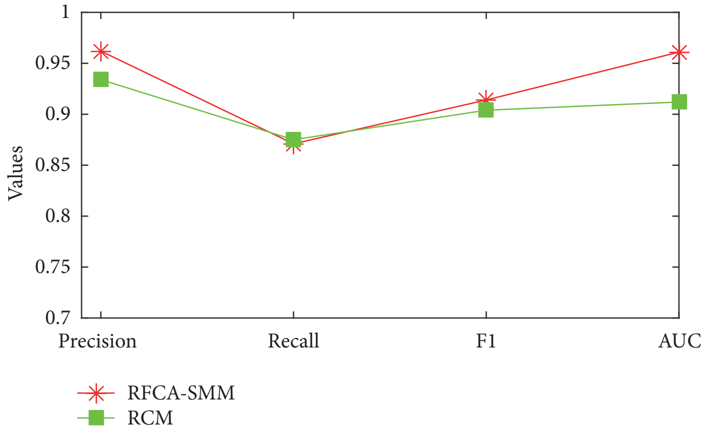


Figure 5: Comparison of RFCA-SMM and RCM results.

show that the proposed algorithm can provide excellent performance with precision rate, F1, and AUC reaching 96.2%, 91.4%, and 96.1%.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant no. 61771185, Grant no. 61772175, and Grant no. 61801171), Science and Technology Research Project of Henan Province (Grant no. 182102210044 and Grant no. 182102210285), Key Scientific Research Program of Henan Higher Education (Grant no. 18A510009), and Postdoctoral Science Foundation of China under Grant 2018M632772.

References

[1] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: connecting heterogeneous social networks with local and global consistency," in *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1485–1494, 2015.

[2] X. Zhou, X. Liang, H. Zhang, and Y. Ma, "Cross-platform identification of anonymous identical users in multiple social media networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 411–424, 2016.

[3] M. M. Mostafa, "More than words: social networks' text mining for consumer brand sentiments," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4241–4251, 2013.

[4] T. Tuna, E. Akbas, A. Aksoy et al., "User characterization for online social networks," *Social Network Analysis and Mining*, vol. 6, no. 1, p. 104, 2016.

[5] J. Liu, F. Zhang, X. Song, Y. Song, C. Lin, and H. Hon, "What's in a name?: an unsupervised approach to link users across communities," in *Proceedings of the the Sixth ACM International Conference*, pp. 495–504, Rome, Italy, 2013.

[6] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 41–49, USA, 2013.

[7] O. Goga, D. Perito, H. Lei, R. Teixeira, and R. Sommer, "Large-scale correlation of accounts across social networks," Tech-nical report, 2013.

[8] K. Cortis, S. Scerri, I. Rivera, and S. Handschuh, "An ontology-based technique for online profile resolution," in *Social Informatics*, Lecture Notes in Computer Science, pp. 284–298, Springer International Publishing, Berlin, Germany, 2013.

[9] P. Jain, P. Kumaraguru, and A. Joshi, "@ i seek "fb. me": Identifying users across multiple online social networks," in *Proceedings of the the 22nd International Conference on World Wide Web Companion*, pp. 1259–1268, Rio de Janeiro, Brazil, 2013.

[10] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, "Exploiting innocuous activity for correlating users across sites," in *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pp. 447–457, 2013.

[11] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the the 22nd ACM international conference (CIKM)*, pp. 179–188, San Francisco, Calif, USA, 2013.

[12] J. Haupt, B. Bender, B. Fabian, and S. Lessmann, "Robust identification of email tracking: a machine learning approach," *European Journal of Operational Research*, vol. 271, no. 1, pp. 341–356, 2018.

[13] Y. Li, Z. Zhang, Y. Peng, H. Yin, and Q. Xu, "Matching user accounts based on user generated content across social networks," *Future Generation Computer Systems*, vol. 83, pp. 104–115, 2018.

[14] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 173–187, IEEE, Berkeley, Calif, USA, 2009.

[15] S. Bartunov, A. Korshunov, S. Park, W. Ryu, and H. Lee, "Joint link-attribute user identity resolution in online social networks," in *Proceedings of the 6th SNA-KDD Workshop*, 2012.

[16] Y. Cui, J. Pei, G. Tang, W.-S. Luk, D. Jiang, and M. Hua, "Finding email correspondents in online social networks," *World Wide Web*, vol. 16, no. 2, pp. 195–218, 2013.

[17] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "HYDRA: large-scale social identity linkage via heterogeneous behavior modeling," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 51–62, USA, 2014.

[18] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," in *Proceedings of the VLDB Endowment*, vol. 7, pp. 377–388, 2014.

[19] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping users across networks by manifold alignment on hypergraph," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, vol. 14, pp. 159–165, 2014.

[20] H. Kobayashi and T. Matsui, "Successful manipulation in stable marriage model with complete preference lists," *IEICE Transaction on Information and Systems*, vol. 92, no. 2, pp. 116–119, 2009.

[21] E. Raad, R. Chbeir, and A. Dipanda, "User profile matching in social networks," in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS)*, pp. 297–304, 2010.

[22] Y. Na, Z. Yinliang, D. Lili, B. Genqing, E. Liu, and G. J. Clapworthy, "User identification based on multiple attribute decision making in social networks," *China Communications*, vol. 10, no. 12, pp. 37–49, 2013.

[23] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause, "Cross-system user modeling and personalization on the social web," *User Modeling and User-Adapted Interaction*, vol. 23, no. 2-3, pp. 169–209, 2013.

[24] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis," in *Proceedings of the the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 225–236, San Jose, Calif, USA, 2016.

[25] K. A. Alam, R. Ahmad, and K. Ko, "Enabling far-edge analytics: performance profiling of frequent pattern mining algorithms," *IEEE Access*, vol. 5, no. 99, pp. 8236–8249, 2017.

[26] W. Cao, Z. Wu, D. Wang, J. Li, and H. Wu, "Automatic user identification method across heterogeneous mobility data sources," in *Proceedings of the 32nd IEEE International Conference on Data Engineering (ICDE)*, pp. 978–989, IEEE, 2016.

[27] T. Hao, J. Zhou, Y. Cheng, L. Huang, and H. Wu, "User identification in cyber-physical space: a case study on mobile query logs and trajectories," in *Proceedings of the the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4, Burlingame, Calif, USA, 2016.

[28] X. Han, L. Wang, S. Xu, G. Liu, and D. Zhao, "Linking social network accounts by modeling user spatiotemporal habits," in *Proceedings of the 15th IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 19–24, IEEE, 2017.

[29] X. L Li, Y. L. Han, D. Y. Zhang, and X. G. Xu, "An evaluation algorithm for importance of dynamic nodes in social networks based on three-dimensional grey relational degree," in *Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 201–212, Springer, Singapore, 2018.

[30] M. Almishari and G. Tsudik, "Exploring linkability of user reviews," in *Computer Security – ESORICS 2012*, vol. 7459 of *Lecture Notes in Computer Science*, pp. 307–324, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2012.

[31] G. Kondrak, D. Marcu, and K. Knight, "Cognates can improve statistical translation models," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 46–48, Edmonton, Canada, 2003.

[32] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[33] H. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian Conference on Computer Vision*, vol. 6493 of *Lecture Notes in Computer Science*, pp. 709–720, Springer, Berlin, Germany, 2010.

[34] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 493–502, 2004.

[35] M. Yan, J. Sang, and C. Xu, "Unified youtube video recommendation via cross-network collaboration," in *Proceedings of the the 5th ACM on International Conference on Multimedia Retrieval*, pp. 19–26, New York, NY, USA, 2015.

[36] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, pp. 1–37, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 5121054, 14 pages
https://doi.org/10.1155/2019/5121054

Research Article

Threat Assessment for Android Environment with Connectivity to IoT Devices from the Perspective of Situational Awareness

Mookyu Park,[1] **Jaehyeok Han**,[1] **Haengrok Oh**,[2] **and Kyungho Lee**[1]

[1] *School of Information Security, Korea University, Seoul 02841, Republic of Korea*
[2] *Agency for Defense Development (ADD), Seoul 05771, Republic of Korea*

Correspondence should be addressed to Kyungho Lee; kevinlee@korea.ac.kr

Received 18 January 2019; Revised 13 March 2019; Accepted 25 March 2019; Published 11 April 2019

Academic Editor: Ilsun You

As smartphones such as mobile devices become popular, malicious attackers are choosing them as targets. The risk of attack is steadily increasing as most people store various personal information such as messages, contacts, and financial information on their smartphones. Particularly, the vulnerabilities of the installed operating systems (e.g., Android, iOS, etc.) are trading at a high price in the black market. In addition, the development of the Internet of Things (IoT) technology has created a hyperconnected society in which various devices are connected to one network. Therefore, the safety of the smartphone is becoming an important factor to remotely control these technologies. A typical attack method that threatens the security of such a smartphone is a method of inducing installation of a malicious application. However, most studies focus on the detection of malicious applications. This study suggests a method to evaluate threats to be installed in the Android OS environment in conjunction with machine learning algorithms. In addition, we present future direction from the cyber threat intelligence perspective and situational awareness, which are the recent issues.

1. Introduction

The Internet of Things (IoT) technology is being applied to various fields such as public safety, transportation, industrial, and healthcare. The expansion of this technology contributes to the quality of human life. The Gartner report predicts that the IoT market will grow at an average annual rate of 28.8 percent from about $ 300 billion in 2015 to more than $ 1 trillion in 2020 [1]. In addition, the report published in 2019 predicts that this IoT technology will be combined with artificial intelligence technologies to develop into "Intelligent Things". With the expansion and evolution of these IoT technologies, many companies are developing application services that can connect their IoT products and smartphone platforms [2]. Typically, Apple developed a "W1" chip to build the IoT ecosystem and began embedding it in IoT products and the iPhone [3]. Samsung is making efforts to install "Tizen", a proprietary operating system, on smartphones, smart cars, and smart home appliances [4]. In addition, Hyundai plans to mass-produce the Near Field Communication (NFC) smart key application in 2019 to control the vehicle through smartphones [5].

The combination of IoT and AI technology is making cyberspace's influence expand to the real world. The cyber-attack in the past occurred only in cyberspace, and the damage was less likely to occur in the environment where it is not directly related to it [6]. However, the damage caused by the cyber-attack is currently spreading from damage to software and hardware, causing a wide range of secondary damage such as leakage of personal information, revision of direction of national policy, and manipulation of public opinion. A typical threat is a mobile malware, such as spyware. Thirty-five percent of attackers use these methods of attack, and the main attack targets include personal and confidential information of high-ranking government officials. In addition, RSA's Current State of Cybercrime, published in 2016, claimed that the mobile fraud targeting financial information in 2015 increased about 173% from 2013 [7].

Global security companies have recently stated that the possibility of a cyber-attack on mobile devices is increasing. They argued that North Korea is concentrating its cyber-attacks on mobile environments that are vulnerable to security. Palo Alto Networks announced that the attack target of malicious application disguised at Google Play Market is Samsung's smartphone users using Korean. McAfee pointed out the "Lazarus", which is supposed to be a hacker organization in North Korea, behind the attacks of malicious Android backdoor applications disguised as biblical applications. In addition, South Korea's security companies are behind a spy app that seized information, text messages, and contact information on about 10 high-ranking officials in the national defense, diplomacy, and security fields in 2016 [8]. The reason for this increase is that personal information or financial information is often stored on smartphones, and smartphones are at the center of connection and control of IoT devices.

Changes in the way in which personal information is stored and the efficient use of IoT equipment are reasons to increase the frequency of attacks on smartphones. These mobile threats can cause damage to both cyberspace and the real world. To provide situational awareness to decision-makers, it is necessary to extend threat detection for a mobile malicious application to evaluate threats. This study proposes a method of assessing threat based on the extracted features through Android malware detection using basic machine learning using the risk model, factor analysis of information risk (FAIR). This paper describes the relationship between mobile threats and cyberspace, threat assessment, and limitation of mobile malware detection in the next sections. In addition, this paper proposes frames from the threat assessment procedure from the perspective of situational awareness section, and the results are described in the result of threat assessment for Android malware application section.

2. The Relationship between Mobile Threats and Cyberspace

With the development of IoT adventure, the border between cyberspace and the real world disappears, and the frequency of cyberattack has a negative effect on the real space. In general, cyberspace is defined as a virtual space that enables communication of the information environment by overcoming the temporal and spatial limits of reality through a virtual network environment composed of electronic devices and electronic spectrum. This environment is interdependent with the Internet, network, embedded processing, human, society, and policy, so the damage from cyber-attacks is likely to expand in the future.

2.1. Mobile Threats to IoT Devices. Mobile devices such as smartphone tablet PCs can be used as access roads to IoT equipment. In recent years, many companies have named IoT devices as smart devices and are controlling them through applications installed on smartphones. That is, if the smartphone is vulnerable to security, data of IoT equipment storing personal privacy information may be leaked. Mobile threats related to these IoT devices can be classified into data leakage, unsecured Wi-Fi, network spoofing, phishing attacks, spyware, etc.

Data Leakage. Mobile applications often cause unintended data leakage. Currently used applications are granted full privileges by the user, but the security of this application is not perfect [9, 10]. They tend to be free of charge in the official App Store. Such information leakage may be prevented through text data. However, about 51.6% of the programs have the risk of information leakage during Android applications, and the attacker can access them through the access token [11]. That is, when a user transmits data to an IoT device through a smartphone, privacy information may be exposed to an attacker.

Unsecured Wi-Fi. In most public places (e.g., airports, parks, etc.), users can use wireless hotspots, but tend to use free Wi-Fi networks. However, these free Wi-Fi generally do not provide secure security. N. Sombatruang et al. have confirmed that personalized photos, e-mail, documents, and login credentials are transmitted without encryption in packets on mobile devices that are connected to unsecured Wi-Fi [12]. If the smartphone is storing data transmitted from IoT devices, it shows that information could be leaked through the fake Wi-Fi.

Network Spoofing. Network spoofing means that an attacker sets up a rogue access point in a public place like a coffee shop, library, or an airport. This method of attack often allows attackers to give users a common name by assigning a common name to access points such as "Free Airport Wi-Fi" or "Coffee House". Many users create an "account" and enter a password to access the free service, at which time users tend to use the same e-mail and password combination for multiple services [13]. This could allow an attacker to expose access to e-mail, e-commerce, and IoT equipment.

Phishing Attacks. Mobile devices are always exposed to phishing attacks because they are almost powered on. Mobile devices are more likely to be a threat than desktops. Even if they receive e-mail, they are less likely to receive warnings through security programs. Such an attack can provide a path for malware to enter the user's mobile device [14, 15]. This may provide the attacker with the rights to the mobile device and the IoT device connected to the device or the personal information stored in the device.

Spyware. Mobile spyware is an application that monitors and records user's privacy and personal information without user's permission. This spyware is installed when a user installs another application, redirects to a malicious website, or physically unlocks the computing device. When a mobile device is infected with spyware, it can eavesdrop on nearby conversations or access data stored or transmitted to the device. For example, if there is a smartphone near the keyboard, it is possible to detect what was entered into the computer through the accelerometer sensor of the smartphone [16]. In addition, an application having a backdoor

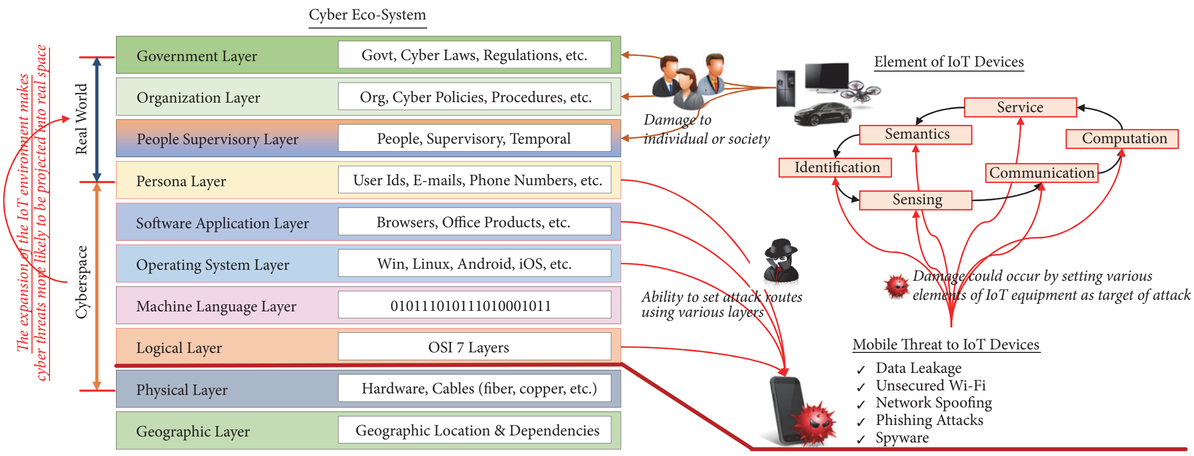

Cyber Eco-System
Government Layer
Govt, Cyber Laws, Regulations, etc.
Organization Layer
Org, Cyber Policies, Procedures, etc.
People Supervisory Layer
People, Supervisory, Temporal
Persona Layer
User Ids, E-mails, Phone Numbers, etc.
Software Application Layer
Browsers, Office Products, etc.
Operating System Layer
Win, Linux, Android, iOS, etc.
Machine Language Layer
0101110101110100010111
Logical Layer
OSI 7 Layers
Physical Layer
Hardware, Cables (fiber, copper, etc.)
Geographic Layer
Geographic Location & Dependencies
Real World
Cyberspace
The expansion of the IoT environment makes cyber threats more likely to be projected into real space
Damage to individual or society
Element of IoT Devices
Service
Semantics
Computation
Identification
Communication
Sensing
Ability to set attack routes using various layers
Damage could occur by setting various elements of IoT equipment as target of attack
Mobile Threat to IoT Devices
✓ Data Leakage
✓ Unsecured Wi-Fi
✓ Network Spoofing
✓ Phishing Attacks
✓ Spyware

Figure 1: Relationship between the element of IoT devices and cyberspace.

may induce the use of push notification services to leak a user's personal information [17].

2.2. Extending the Mobile Threats to the Real World with IoT Devices. The mobile threats can affect not only mobile devices, but also people using IoT equipment and IoT technology. In other words, it means that the mobile device acts as a medium that connects cyberspace and real space. This concept can be explained through the cyberspace layer described in US Joint Publication (JP) 3-12R Cyber Operation. According to this document, cyberspace is composed of a physical layer, a logical layer, and a persona layer. The physical layer is a layer where physical devices such as IoT devices, routers, switches, and mobile devices are geographically and physically. The logical layer is a layer where network or communication devices are located in the physical layer. The persona layer is a layer that represents a human being in a virtual space through an electronic service provided at the physical layer and logical layer and corresponds to an IP address, an e-mail, and an ID [18].

These cyberspace layers are subdivided and expanded into the basic concepts of physical layer, logical layer, and persona layer. David Clark subdivided cyberspace into a physical and a logical layer, as well as an information layer that represents the information flow in cyberspace and a top layer-people layer that allows human decisions to be projected into cyberspace. The characteristic of this concept is that the service of cyberspace has a dependency between the layers, thus providing a basis for the threat of cyberspace to be reflected in the real space [19]. The US Department of Homeland Security has defined a cyber ecosystem that extends the concept of the cyberspace layer. The cyber ecosystem is a notion that social organizations such as private companies and governments as well as human beings construct virtual ecosystems by interacting with electronic devices and electronic services such as hardware and software [20]. This concept reflects the situation in which the gap between cyberspace and real space is decreasing due to the development of IoT and AI (Artificial Intelligence) technology. The cyber ecosystem consists of 15 layers including the cyberspace layer concept. These hierarchies suggested that the impacts in cyberspace could affect social or policy systems (see Figure 1).

This means that the threat of cyberspace through mobile devices can be visualized through IoT equipment. The reason why IoT equipment can be visualized as a threat is due to the factors of IoT. IoT equipment consists of identification, sensing, communication, computation, service, and semantics [21].

Identification. It is an explicit identification of each object in the logical layer and is provided by addressing the device, such as an electronic product code (EPC) and ubiquitous code, by naming the device or by addressing a specific object. IPv6 is used to assign a unique address to each entity [22]. Mobile devices could be exposed to threats if they are made into fake access points with the name of such equipment.

Sensing. It refers to the process of collecting information from objects through RFID tags, smart sensors, and so on. Since most IoT devices are connected to the mobile device, if the mobile device is exposed to spyware, phishing attacks, etc., an attacker can gain access to private data [23]. Or when a user accesses unsecured Wi-Fi, the ID and password of the IoT device may be exposed and illegally monitored or the personal information may be leaked to the outside.

Communication. It is one of the main purposes of IoT technology because various IoT devices are connected to each other. This element can send and receive messages, files, and other information stored on IoT devices and utilize communication technologies such as Near Field Communication (NFC), Bluetooth, Wi-Fi, and Long-Term Evolution (LTE) [24]. However, if mobile devices connected to communication are exposed to threats such as spyware, there is a possibility that private information of IoT equipment articles connected to mobile devices may be exposed.

Computation. It is an element that uses sensors to perform calculations on information collected from objects and is developed to perform processing in IoT applications. Typically, this is the operating system. Mobile devices such as smartphones are divided into Android and iOS. The vulnerabilities of OS are used for spyware or malware. In other words, when a smartphone with an IoT application installed on a smartphone is attacked, IoT devices may also be exposed to threats [25].

Service. It is provided by IoT applications and can be divided into four types. The identity-related service is used to get the ID of the object that sent the request. An information aggregation service aims to collect all information from an object. The collaboration service makes decisions based on the information gathered and sends appropriate responses to the devices. Finally, ubiquitous services are used to respond to devices immediately, regardless of time and location [26]. However, malicious applications such as these can be used for data leakage and user surveillance.

Semantics. It is an element that reduces the gap between cyberspace and real space, ensuring the convenience of users in data collection and utilization of IoT equipment. This element is responsible for gathering information from IoT equipment and making appropriate decisions to send a response to the mobile device [27]. Since the decision-making power is on the human side, the mobile device is used to remotely control it, but if the mobile device itself is exposed to threats, semantics can be exploited by an attacker.

3. Threat Assessment

To respond to an attack, the manager performs a risk assessment on the assets he holds. The risk assessment consists of threats, vulnerabilities, assets, and missions (or countermeasure). If the assets to be protected are clear, the administrator will focus on the threats associated with them. Recently, Cyber Threat Intelligence (CTI) has been used to counter APT attacks. However, in the case of a threat, an attacker needs to organize the threat into tactics, techniques, and procedures (TTP) because the attacker sets the attack method through several different cases. This section describes the relevance of threat assessment and situational awareness (SA) for evaluating cyber threat information and the Factors Analysis of Information Risk (FAIR) model applied in this study.

3.1. Threat Assessment in Situational Awareness. Situational awareness (SA) is the process of recognizing an environmental element in a threat situation or the time and space where a specific event occurs and establishing countermeasures. These procedures are used as a framework to recognize, judge, and respond to threats such as terrorism, security, and cyber security. The SA is used in command and control systems that are used for decision support in the military sector.

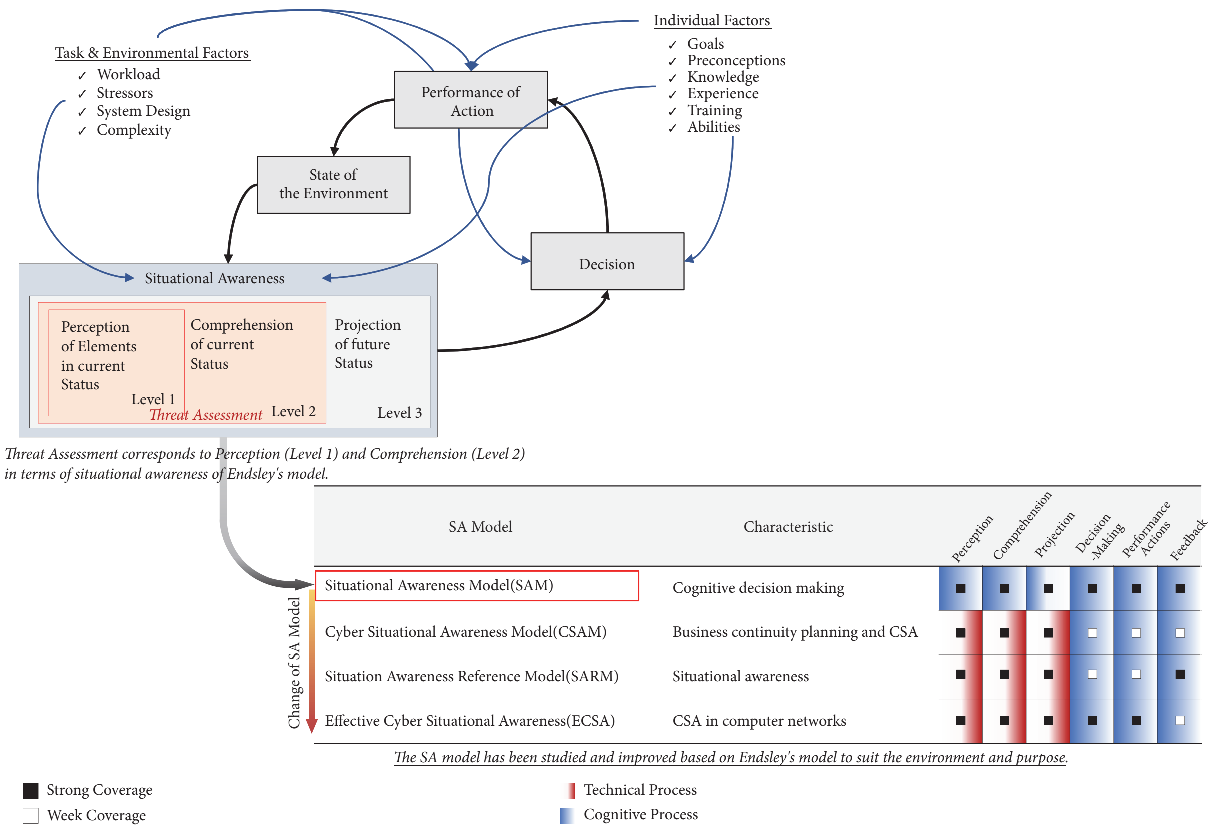
The basic concept of SA extends from Endsley's model. The Endsley's model is a process for understanding the perceptions of the environment, understanding the changes in the current situation, and foreseeing the consequences of future projections. The Endsley's model has three phases of perception, comprehension, and projection. "Perception" recognizes the status and attributes of related elements in the current environment. "Comprehension" recognizes the current situation through elements collected at the "Perception". "Projection" assesses how the information analyzed in "Comprehension" will affect the state of the future operating environment (see Figure 2) [28].

In cyberspace, the SA model is being developed through the basic concept of this Endsley's model. J. Okolica et al. proposed a cyber situational awareness model (CSAM) with business continuity planning (BCM). This model updates cyberspace assets or systems in real time and predicts future threats through sense, evaluate, and assess [30]. G.P. Tadda and J.S.Salerno developed the Situational Awareness Reference Model (SARM) to improve understanding of various data in cyberspace. The SARM can actively respond to changing threats in real time [31]. N. Evancich et al. proposed Effective Cyber Situational Awareness (ECSA) focusing on network security. ECSA is divided into three stages: "Network Awareness", "Threat Awareness", and "Operational Awareness". "Network Awareness" is a step where a decision maker recognizes the characteristics of the network's assets and security. The "Threat Awareness" step detects an attack vector that has entered the network. "Operational Awareness" is a measure of damage to network operation capability due to the threat. ECSA is an improved situational awareness model than CSAM in decision-making, collaboration, and resource management [32].

The SA models, which are being studied recently, seek to recognize changes in the current situation by detecting threats. From the viewpoint of SA, threat assessment is a link between policy and technology to utilize technologies such as the prediction of a future threat, prediction of attack path, and identification of countermeasure, as well as threat detection in the perception of awareness stage [33]. Threat assessment in cyberspace is an engineering methodology that detects, identifies, and prioritizes cyber threats in order to apply countermeasures that reduce vulnerabilities to cyber-attacks. Many kinds of research and developments have been made in the evaluation of cyber threats. The CyberPrep Working Group has established cyber-aware enterprise transformation strategies that reflect the understanding of APT attacks, organizational responses, and cybersecurity investment strategies. MITER's center for resiliency experimentation developed cyber resiliency engineering to develop a methodology for processes, personnel, and individual systems that support resilience strategies and techniques for mission functions against cyber threats. In addition, this threat assessment uses System/Acquisition Mission Assurance Engineering (SAMAE) applying System Development Lifecycle (SDLC) to analyze APT attack knowledge [34]. This threat assessment includes the following elements in common.

(i) Identify and prioritize high-risk tactics, techniques, and procedures (TTP) that cyber-assets can be affected

(ii) Identify and prioritize effective countermeasures against identified TTP.


Task & Environmental Factors
✓ Workload
✓ Stressors
✓ System Design
✓ Complexity
Individual Factors
✓ Goals
✓ Preconceptions
✓ Knowledge
✓ Experience
✓ Training
✓ Abilities
Performance of Action
State of the Environment
Decision
Situational Awareness
Perception of Elements in current Status
Level 1
Comprehension of current Status
Level 2
Projection of future Status
Level 3
Threat Assessment
Threat Assessment corresponds to Perception (Level 1) and Comprehension (Level 2) in terms of situational awareness of Endsley's model.
Change of SA Model
The SA model has been studied and improved based on Endsley's model to suit the environment and purpose.
■ Strong Coverage
□ Week Coverage
Technical Process
Cognitive Process

SA Model	Characteristic	Perception	Comprehension	Projection	Decision -Making	Performance Actions	Feedback
Situational Awareness Model(SAM)	Cognitive decision making	■	■	■	■	■	■
Cyber Situational Awareness Model(CSAM)	Business continuity planning and CSA	■	■	■	□	□	□
Situation Awareness Reference Model(SARM)	Situational awareness	■	■	■	□	□	■
Effective Cyber Situational Awareness(ECSA)	CSA in computer networks	■	■	■	■	■	□

FIGURE 2: Change and development of situational awareness model [29].

(iii) Recommend a countermeasure to reduce the possibility of cyber asset attacks

There are limitations in the risk assessment method of measuring the risk by setting the range of the cyber asset in the situation where the boundary between cyberspace and the real space is unclear such as IoT environment. Since the IoT environment provides a diverse attack path, the threat assessment can contribute to improving the risk management approach to assets [35].

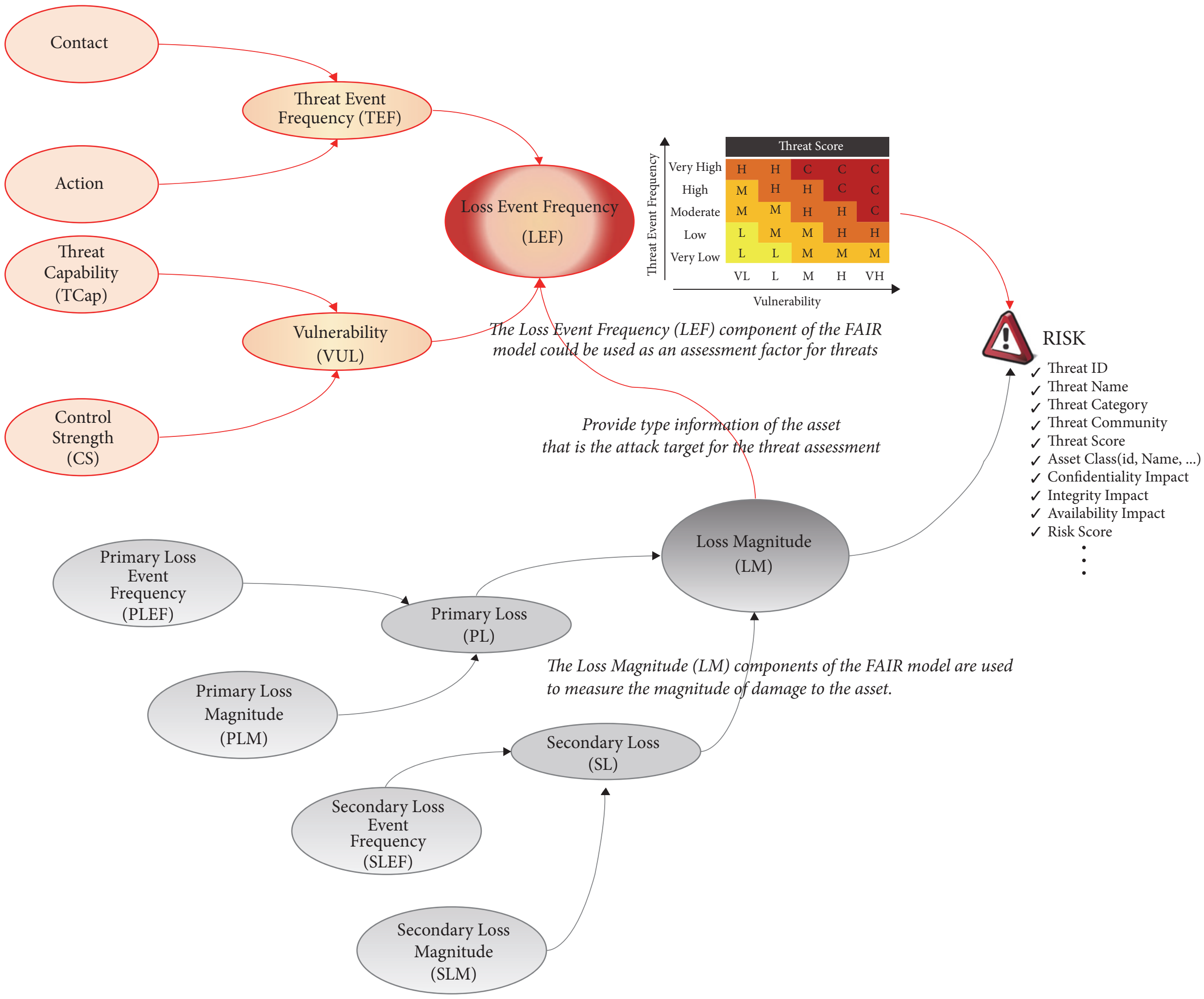
3.2. FAIR Model for Threat Assessment.

The factor analysis of information risk (FAIR) model is a risk management model developed by J.A. Jones. This model focuses on the probability of an emerging threat event and measures threat and asset by frequency and size to measure risk. This FAIR model measures the risk by combining Loss Event Frequency (LEF) and Loss Magnitude (LM) (see Figure 3). This model reflects in detail the probabilities and probabilities. The likelihood is expressed as 100% or 0% when a threat condition is possible in a binary condition, and the probability reflects continuity between absolute certainty and impossibility. By using this probabilistic approach, it is possible to balance the compensation probability with the understanding of the loss probability, and it has an advantage that the range of the acceptable level of the decision maker (or manager) can be quantified with respect to the risk level.

In this model, the part that can be utilized as a frame of threat assessment is LEF. The LEF is the frequency with which the threat agent is likely to be harmed by the threat agent in a certain period or situation and consists of Threat Event Frequency (TEF) and Vulnerability (VUL) [36].

Threat Event Frequency (TEF). It is the frequency with which a threat agent is likely to act on an asset within a specified period, although the threat agent may act on the asset, but not on the success of the attack. A typical example is a hacker who has not successfully attacked a web server. Such an attack is not a loss event, but can be considered a threat event. This TEF consists of *Contact* and *Action*, and the attacking action is based on the contact of the threat agent. *Contact* refers to the frequency with which a threat agent may contact an asset within a certain period. Types of contact are classified as “Random”, “Regular”, and “Intentional”. *Action* refers to the probability that a threat agent will perform an actual attack on an asset in a situation where a contact of the threat agent occurs. The preconditions for the action are intelligent threats such as “Thinking” threat agents and malicious programs.

Vulnerability (VUL). It means the probability that an asset is unable to resist the action of the threat agent. The vulnerability occurs because there is a difference between the power of the threat agent and the ability of the asset to resist it. That is, the vulnerability is relative to the type of threat. These VULs are measured as a combination of *Threat Capability*



Threat Event Frequency	Threat Score				
Very High	H	H	C	C	C
High	M	H	H	C	C
Moderate	M	M	H	H	C
Low	L	M	M	H	H
Very Low	L	L	M	M	M
	VL	L	M	H	VH

The Loss Event Frequency (LEF) component of the FAIR model could be used as an assessment factor for threats

Provide type information of the asset that is the attack target for the threat assessment

The Loss Magnitude (LM) components of the FAIR model are used to measure the magnitude of damage to the asset.

Figure 3: Using the FAIR model for threat assessment.

(TCap) and *Control Strength (CS)*. *TCap* means the level of expected threat agent power that could have a negative impact on the asset. Since all the threats of *TCap* are not created the same, the threat agent does not perform the same function in one threat community. Also, although the *TCap* may be high for an attack target in which a threat agent is set, it may be incompetent for other objects. *CS* refers to the strength with which an asset possesses resistance against threats, measured against the threat agent or threat community. If the *CS* is set to a small number of controls, the probability value of each control can be calculated independently.

The elements that measure these threats can create a profiling list considering the assets and threat agents that can be the targets of the attack at the end. The attackers are called the *threat community (TCom)*, which is divided into Nation-States, Groups, and Individuals according to the "NIST SP 800-30: Risk Management Guide for Information Technology Systems". Nation-States refers to threat agents who conduct cyber-attacks by government or government support. Typical examples are APT37 (North Korea), APT32 (Vietnam), and APT33 (Iran). Group refers to threat agents for political ideals or legitimate and illegal gains against cyber-attacks, such as Anonymous, APT18 (Wekby), APT19 (Codoso), and APT28 (Tsar). These organizations receive informal support, so if the sponsor organization becomes a state or government by cyber threat TTP in the future, the threat community may be changed to Nation-States. Individuals consist of outsiders, insiders, etc. and perform cyber-attacks based on personal beliefs or retaliation. A typical example is Edward Snowden. Through the classification of *TCom*, threat profiling stores and manages attack patterns and characteristics of malicious attackers with motive, primary intent, sponsorship, preferred general target characteristics, preferred targets, capability, personal risk tolerance, and concern for collateral damage.

4. Limitation of Mobile Malware Detection

Research and development for the detection of malicious behavior by attackers are one of the important technologies in the field of security. As the usability of mobile device increases in IoT environment, threatening behavior is changing not only damage to one device, but also damage to the whole environment of the society. From this point of view, most malicious behavior detection studies tend to focus on automation technology such as training set configuration and detection algorithm, such as machine learning [37].

Z. Aung and W. Zaw proposed a semisupervised algorithm to detect malware in Android. This research extracted permission features from Android's *apk* files and clustered files suspected of malicious activity. In addition, this research classified clusters of malware using three methods: Decision Tree Algorithm, Random Forests (RF), Classification and Regression Tree (CART). They found that for 500 sample Android applications, the RF algorithm showed a high accuracy of 91.8% [38]. D. J. Wu et al. proposed a static analysis-based mechanism to detect Android malware. The proposed model clusters the information related to permissions and intents in the manifest file by setting them as features. Finally, this research applied the classification algorithm to detect Android malware. They experimented with Expectation Maximization (EM), clustering method of K-means algorithm, k-nearest neighbor (*k*-NN) and Naïve Bayes classification algorithm and found that the algorithm combining K-means and KNN is 97.87% [39]. N. Khan et al. proposed an efficient method for detecting malicious Java scripts in web applications. In this research, feature subset uses the wrapper method to reduce dimensions and supervised learning algorithms to achieve high accuracy. They applied support vector machine (SVM), Naïve Bayes, decision tree, *k*-NN, and RF algorithms and found that the *k*-NN algorithm achieves about 98.3% accuracy [40].

H. S. Ham et al. conducted research to detect Android malware from the viewpoint of a secondary impact such as privacy infringement and information leakage caused by combination of IoT and smartphone. They used a linear SVM algorithm to detect Android malware. As a result, this study showed that the linear SVM algorithm yielded 95.7% higher than other machine learning classification algorithms (average precision = 68.7%) [41]. J. Sahs and L. Khan detected Android malware using a one-class SVM algorithm. They used the permissions of the application as a feature, resulting in about 50% true negative (TN) and about 90% true positive (TP) for 91 malicious Android applications out of 2081 [42]. M. G. Schultz et al. studied the malware detection of Android malware as a feature of the DLL and the raw hexadecimal representation of the system call, string, and binary generated by the program. This study compared the accuracy of the signature method, RIPPER, Naïve Bayes, and Multi-Naïve Bayes algorithm. As a result, the Multi-Naïve Bayes algorithm showed a higher accuracy of 97.76% than the other algorithms [43].

K. Riad and L. Ke detected malicious applications in the Google Play Store. This research proposes a RoughDroid algorithm, a floppy analysis technology that can discover Android malware applications directly on smartphones. They extracted the features of the XML *manifest* file and *Dex* file of the Android application as features and obtained the accuracy of 95.6% [44]. I. Martin et al. analyzed indirect functions and metadata to identify patterns in Android malware applications. The research focused on malware detection, such as application developers and certificate issuers published on the Android Market. They used logistic regression (LR), SVM, and RF to construct a small, efficient classifier that could detect malware applications early in the sandbox [45]. W. Niu et al. proposed a method to detect the advanced persistent threat (APT) malicious code command and control (C and C) domain with high accuracy by analyzing the mobile DNS log. The study scored the domain through Alexa rankings and VirusTotal and identified the C and C domain of malware using the Global Abnormal Forest (GAF) algorithm. As a result, they confirmed that the proposed method achieves more than 99% accuracy compared with the local outlier factor (LOF), *k*-NN, and iForest algorithm [46].

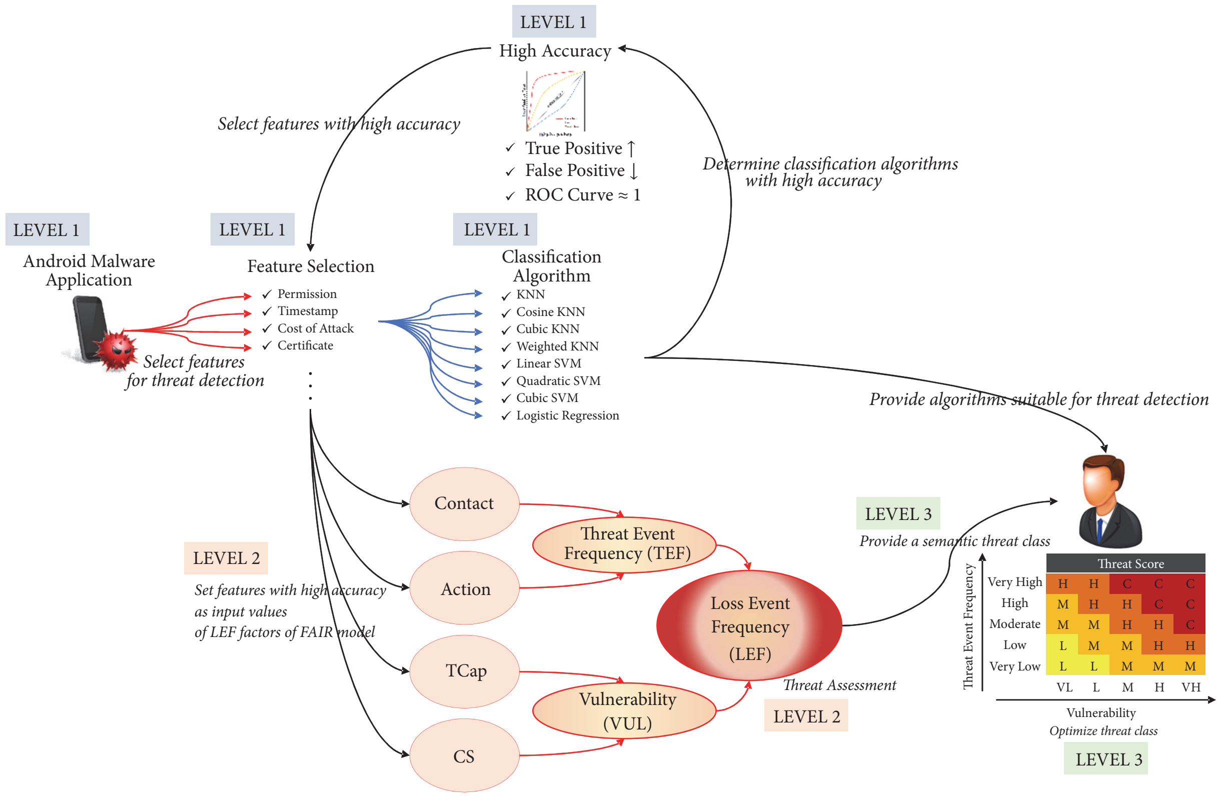
Most studies are focused on improving the effectiveness or accuracy of malware detection. However, in recent years, the focus of CTI on security has focused on preventing accidents. Detection of malicious behavior of an attacker is also an important technology, but whether or not the detected result can express a criterion to respond to a threat is one of the important challenges. Recently, MITRE has emphasized that it is the task of the current security field to recognize threats while presenting the method of Threat Assessment and Remediation Analysis (TARA) [34]. In addition, Fireeye, a global security firm, publishes intelligence reports quarterly and claims that measurement and evaluation of cyber threats are essential [47].

5. Threat Assessment Procedure from the Perspective of SA

In terms of the situational awareness model, threat measurement tends to depend on the human cognitive judgment. This study measures the *LEF* factors of the FAIR model as a probability distribution to minimize qualitative judgments and provide objective indicators to decision-makers. These probability distributions aim at securing objectivity of threat assessment by detecting actual malware data. The proposed SA model consists of three levels. The first level is *Malware Awareness*, which detects the attacker's malicious behavior using supervised learning. The second level is *Threat Awareness*, which maps the features of detected malicious activity to the LEF factors of the FAIR model and then evaluates the threat. This level also clusters the threats to the number of classes required by the decision maker. The final level is *Decision-Making Awareness*, which provides decision-makers with an optimal level of threat (see Figure 4) [48].

5.1. Malware Awareness.

As the expansion of the IoT environment increases the usability of mobile devices, cyberattacks on smartphones are increasing. Particularly, this is done by a malicious application of attack on the mobile environment, and most attacks are an unknown attack. However, existing rule-based algorithms have limitations in actively detecting new threats (e.g., unknown malicious applications). The *Malware Awareness* Level aims to detect threats using machine learning. The features of the high detection result (e.g., ROC, True Positive, Accuracy, etc.) obtained through the classification of the threat are used as



Threat Score	VL	L	M	H	VH
Very High	H	H	C	C	C
High	M	H	H	C	C
Moderate	M	M	H	H	C
Low	L	M	M	H	H
Very Low	L	L	M	M	M

FIGURE 4: Situational awareness model for measuring threat using Android malware detection.

the probability distribution of the FAIR model's *LEF* factors for threat measurement.

5.2. Threat Awareness. The probability distribution of features that resulted in high detection results at the *Malware Awareness* Level becomes the input value of the *LEF* factors. This study matches the detected results to *TEF* and *VUL* (*TCAP, CS*). There is a limit to the difficulty in clearly distinguishing the threat class from the *LEF* measured by the joint probability of *TEF* and *VUL*. To solve this problem, this research uses the K-means clustering algorithm. Based on the matched results, the decision maker clusters n features into K threat classes with D dimensions. For example, in the case of five ratings from "Very High" to "Very Low", K is *5*. The probability value of the features matched to *TEF* and *VUL* is f_n. In this case, $r_{nk} = 1$ if the n th f_n corresponds to k $(k = 1, 2, 3, \cdots, K)$ th threat cluster, and *0* is otherwise. The distortion measure function for clustering the threat class is *TC*. This can be expressed as [49]

$$r_{nk} = \begin{cases} 1, & \text{if } k = \underset{k}{\operatorname{argmin}} \left\| f_n - \mu_k \right\|^2 \\ 0, & \text{Otherwise} \end{cases} \tag{1}$$

$$TC = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| f_n - \mu_k \right\|^2 \tag{2}$$

To minimize the threat class function *TC* value, we use the iterative method EM (Expectation Maximization) algorithm. The EM algorithm is a process of iteratively fixing one of μ_k and r_{nk}, and the convergence value is calculated through it. The *TC* value is the minimum value for m. Therefore, if the function *TC* is differentiated to μ_k, μ_k is used as a value to cluster the threat class. This can be expressed as [50]

$$\mu_k = \frac{\sum_n r_{nk} f_n}{\sum_n r_{nk}} \tag{3}$$

5.3. Decision-Making Awareness. Even if the threat class is clustered through the K-means algorithm at the *Threat Awareness* Level, there may be a limit to the semantic decision. To overcome these limitations, this level is optimized by clustering threat class using Gaussian Mixture Model (GMM). In this paper, we assume that the distributions of the threat class are combined into the number of K Gaussian distributions. The GMM can statistically deduce the characteristics of these K Gaussian probability distributions. The value of K is the number of threat classes set at *Threat Awareness* Level. The GMM performance measure is a log likelihood function that estimates the parameter maximizing the probability of the *TC* function at the *Threat Awareness* Level. The parameter *TC* of the log likelihood function, which optimizes the threat at the *Decision-Making Awareness* level, is the mean μ_k, covariance Σ_k of the Gaussian distributions and the probability π_k. This is equivalent to

$$\ln p\left(TC, \pi, \mu, \Sigma\right) = \sum_{n} \ln \sum_{k} N\left(TC_n \mid \mu_k, \Sigma_k\right) \tag{4}$$

Since the GMM optimization algorithm is also difficult to estimate by jointly updating, this level uses the EM algorithm, which is an alternative update method. This *Decision-Making Awareness* level minimizes the log likelihood function to provide an optimized threat class for decision-makers. It fixes π_k, calculates μ_k, Σ_k, or fixes μ_k, Σ_k and computes π_k. Equations to optimize the threat class are 5, 6, and 7. z is a parameter that makes optimization calculations easier with latent variables [51, 52].

$$\mu_k = \frac{\sum_n p\left(z_{nk} = 1 \mid TC\right) TC_n}{\sum_n p\left(z_k = 1 \mid TC\right)} \quad (5)$$

$$\Sigma_k = \frac{\sum_n p\left(z_{nk} = 1 \mid TC\right)\left(TC_n - \mu_k\right)\left(TC_n - \mu_k\right)^T}{\sum_n p\left(z_k = 1 \mid TC\right)} \quad (6)$$

$$\pi_k = \frac{\sum_n p\left(z_{nk} = 1 \mid TC\right)}{n} \quad (7)$$

6. Result of Threat Assessment for Android Malware Application

To maximize the availability of IoT devices, companies developing IoT devices are encouraging applications to be installed on user's smartphones to help connect their IoT devices. However, when a users' smartphone is exposed to a malicious application, the personal information stored on the user's smartphone and the privacy information collected through the IoT device may be leaked to the attacker. Therefore, when detecting the malicious behavior of the smartphone in the center of the multiple link system, it is important to the process of assessment by CTI as well as the process of producing it. This section measures the threat of a malicious application in the Android environment by applying the threat assessment procedure of the SA model perspective proposed above.

6.1. Detection of Android Malicious Application. The data of this study refer to the malicious application data obtained by static analysis from J. Jang et al. This data consists of APIs related to malicious behavior of Android applications, and a list of system commands. The total number of data is 2000 (1500 normal applications, 500 malicious applications) [53]. Based on this data, this study applied classification algorithms (e.g., KNN, SVM, Logistic Regression, etc.) with high accuracy during related studies. The Confusion Matrix was used for the measured results.

k-Nearest Neighbor (k-NN). It is a nonparametric method used for classification or regression. This algorithm consists of the nearest training data in the feature space with k. In k-NN, objects are classified by majority vote of objects assigned to the most common items among nearest neighbors of k. This study utilized various k-NN algorithms. The basic k-NN algorithm utilizes the Euclidean distance measure. Cosine k-NN is a method of using cosine similarity to cluster the similarity between vectors measured using the cosine of the angle between two vectors of inner space. Cubic k-NN is a clustering method using Minkowski space distance measurement. If this space is used, it is effective, for high-dimensional calculation because it has the advantage of expressing more than four dimensions. Weighted k-NN is a method of clustering by assigning the distance weight as a reciprocal of the square. This method has been used for learning data near distance for new input data has more influence on the decision than neighboring data far away. [54].

Support Vector Machine (SVM). It is a method of classifying through a set of hyperplanes. This algorithm classifies the hyperplane into data that makes the classifier error small. That is, the categorized application data calculates the hyperplane with the greatest distance from the nearest malicious application data. To calculate this, the algorithm classifies the data by defining a kernel function $k(x, y)$. The sum of $k(x, y)$ represents the degree of proximity of the training data x and its corresponding data point x_i and measures the relative proximity of these points. SVM uses a linear (1st order kernel function) as a technique to find the hyperplane that maximizes the margin while classifying the data well. However, since it is difficult to classify data completely through any straight line, various kernel functions should be used for accurate classification. In this study, kernel functions were modified from first to third order, and kernel functions were transformed into Gaussian functions [55].

Logistic Regression. It refers to the use of the relationship between dependent and independent variables as a concrete function for future prediction models. This algorithm is a classification technique in which the dependent variable is categorical data and the result of the data is classified into a specific class when the input data is given. This study classified malicious application features into categorical data [56].

The *Malware Awareness* of this study derives its results through the Confusion Matrix, which evaluates the performance of machine learning algorithms. The Confusion Matrix is represented by true positive (TP), false negative (FN), false positive (FP), and true negative (TN). TP is a measure of normal activity of an application as normal behavior. FN is a measure of normal behavior as malicious behavior. FP is a measure of malicious behavior as normal behavior. TN is an indicator of malicious behavior as malicious behavior. Through these indicators, true positive rate (TPR), positive predictive value (PPV), false negative rate (FNR), accuracy (ACC) and $F1$-score can be measured (see from (8) to (12)) [57].

$$\mathrm{TPR} = \frac{TP}{TP + FN} : \textit{Sensitivity rate} \quad (8)$$

$$\mathrm{PPV} = \frac{TP}{TP + FP} : \textit{Precision} \quad (9)$$

$$\mathrm{FNR} = \frac{FN}{FN + TP} : \textit{Miss rate} \quad (10)$$

$$\mathrm{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

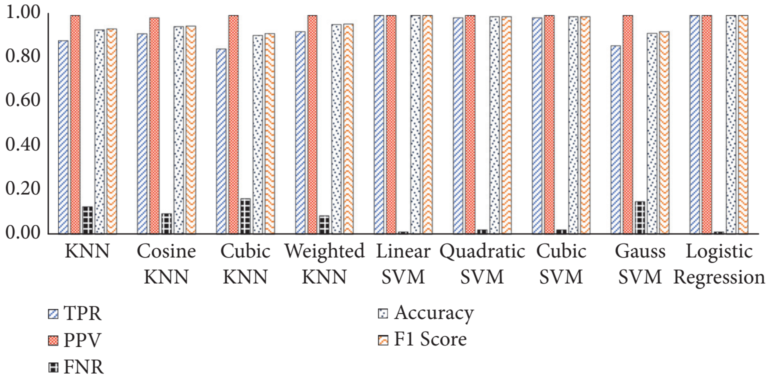


Figure 5: Detection results of Android malware using machine learning.

Table 1: Detection results using k-NN, SVM, and Logistic Regression Algorithms.

Measure	k-NN	Cosine k-NN	Cubic k-NN	Weighted k-NN	Linear SVM	Quadratic SVM	Cubic SVM	Gauss SVM	LR
TPR	0.876	0.907	0.839	0.917	0.990	0.980	0.980	0.853	0.990
PPV	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
FNR	0.124	0.093	0.161	0.083	0.010	0.020	0.020	0.147	0.010
ACC	0.925	0.940	0.900	0.950	0.990	0.985	0.985	0.910	0.990
F1	0.930	0.942	0.908	0.952	0.990	0.985	0.985	0.917	0.990

$$F1 - \text{Score} = \frac{2TP}{2TP + FP + FN} : \quad \text{Harmonic mean of precision and sensitivity} \tag{12}$$

The Android malicious application is categorized into 136 features corresponding to permission and intent, resulting in high accuracy (see Table 1; Figure 5). In particular, the algorithms that show the highest outcome among the algorithms used are SVM series, such as linear SVM, Quadratic SVM, Cubic SVM, and logistic regression. The features (such as "permission" and "intent") that show high detection results are used in *Threat Awareness* as factors of threat in FAIR model.

6.2. Threat Assessment of Android Malicious Application. Detection of malicious Android applications can achieve accurate results using previously studied machine learning algorithms. However, in the situational awareness (or decision maker) viewpoint, there is a limit in choosing the countermeasure through the detection of the threat. In other words, decision-makers must consider the cost of ignoring or responding to threats to assets they hold. To maximize the effective use of machine learning, the threat assessment method proposed in this study utilized the factors of *LEF* of the FAIR model.

Threat Event Frequency (TEF) of Malicious Application. The *TEF* of the FAIR model consists of the factors of *Contact* and *Action*. *Contact* indicates the possibility of an attack, usually measured in frequency. *Action* means that the attacking action works in a real environment, and it is mainly expressed by probability. In other words, the combination of these two factors is expressed as the frequency of malicious application's threats in the Android environment. However, the frequency of these threats is not an indication of the success of the attack. Previously, the features used for detection at the *Malware Awareness Level* were identified as "permission" and "intent". The features of "permission" have features that allow access to most data, such as camera, location, call history, etc. in the Android environment. Typical examples are *android.permission.GET_ACCOUNTS*, *android.permission.BLUETOOTH_ADMIN*, *android.permission.READ_SMS*, etc. In the case of "intent", most of the less relevant features (*android.intent.action.MAIN*, *android.intent.action.BOOT_COMPLETED*, *android.intent.action.DATA_SMS_RECEIVED*, etc.) that access malicious application were excluded from the threat assessment process. There is a research result that features of "permission" act as a features of the malicious applications as mentioned in *Limitation of Mobile Malware Detection*. Based on the detection results, the *TEF* is set to the probability distribution of the features of the "permission" series and reflected in the threat assessment.

Vulnerability (VUL) of Malicious Application. *VUL* is caused by the difference between the power of threat (*TCap, Threat Capability*) and the defending ability (*CS, Control Strength*) of the target asset. The threat in this process is whether the application installed in the Android environment has access to the device's data. Therefore, *TCap* can be configured with the probability of "permission" features included in the Android malicious application. In other words, the more access privileges a malicious application has, the more robust it is. These threat capabilities have various malicious "permissions" that an attacker could illegally acquire various

Table 2: Utilizing "Zerodium Payouts for Mobiles" for Index of *Control Strength (CS)*.

Range of Price	Android Vulnerabilities	Price Measure
Up to $500,000	Email App RCE+LPE	9.5
	SMS/MMS RCE+LPE	9.18
	Signal RCE+LPE	8.22
	Viber RCE+LPE	7.58
Up to $1,500,00	Chrome RCE+LPE	6.94
	Documents RCE+LPE	6.62
	Media Files RCE+LPE	6.3
	Baseband RCE+LPE	5.98
Up to $100,000	LPE to Kernel	5.34
	Wifi RCE+LPE	4.7
Up to $50,000	Chrome UXSS/SOP	4.38
	LPE to Root	4.04
	RCE via MitM	3.72

pieces of information. *CS* is the defensive ability of assets. This is the same as the ability to find vulnerabilities in the Android environment and to test them against an exploit kit. These indices can be set based on the price traded in Android malware. "Zerodium" is a representative company that deals with mobile malware. Through "Zerodium Payouts for Mobiles", this study sets up CS for the Android OS (see Table 2) [58].

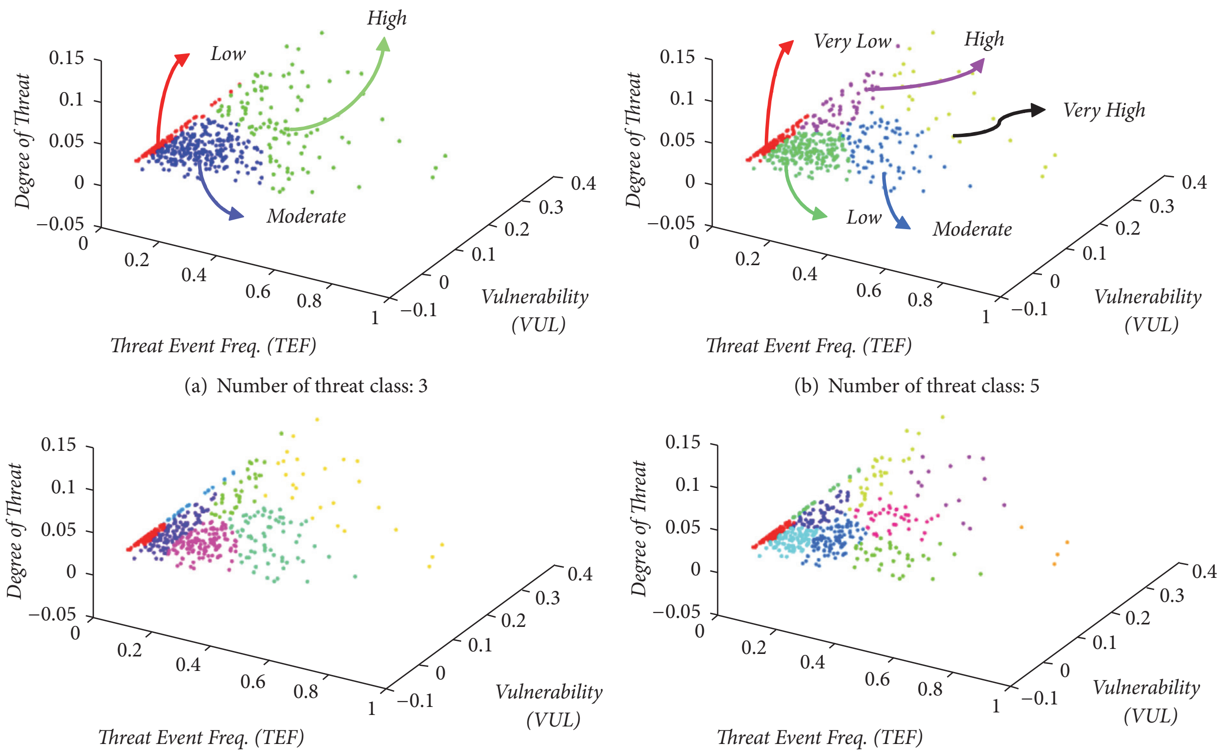
For the threat assessment, this study assumed that each probability distribution of *TEF* and *VUL* for the Android Malicious application was generated under independent conditions. Through this, not only the *TEF* and *VUL* factors, but also the threat class (*LEF*) can be measured. In the case of such a combination, it is difficult to intuitively judge the current situation from the viewpoint of the decision maker. For this reason, the above-mentioned method adds an optimization step through GMM. That is, at the Threat Awareness Level, the decision maker can classify three (*Low, Moderate, and High*), five (*Very Low, Low, Moderate, High, and Very High*), or overall seven classes by changing the k value against the threat, and visualization is possible through the optimization process through GMM. The advantage of subdividing this threat class is the ability to determine the time and cost of investing in countermeasures to counter the threat (see Figure 6).

A representative example from the point of view of threat assessment is TARA. In this case, the basic strategy is to measure threats and identify sophisticated countermeasures. In particular, this method assumes that each countermeasure is independent when selecting a countermeasure. This has the limitation that the selected countermeasures can reduce the efficiency of the other countermeasures. However, if the elements of the threat are identified and probabilistic, such as the approach presented in this study, a conditional approach to the countermeasure of the threat is possible. In addition, when combining with recent threat detection technology using machine learning, deep learning, artificial intelligence, etc., it is possible to contribute to the efficiency of decision-making by securing the real time of threat evaluation.

7. Summary and Discussion

The expansion of the IoT environment is making connectivity between cyberspace and real space stronger. Also, for efficient control of the IoT environment, many companies are concentrating on developing mobile devices such as smartphones (hardware development). However, the security of the application installed on the smartphone is weak. This results in a higher attack rate of malicious attackers. The malicious application is being updated on the market (e.g., Google Play Store) without being verified by the administrator. Many studies have focused on the detection of these malicious applications, and their accuracy and efficiency are approaching commercialization. However, there is a limit to utilize the detected result of a malicious application for decision-making from the manager's point of view. Also, existing risk assessment studies are concentrated on owning assets, so there are limitations that simplify the threat, and there are few studies evaluating threats in connection with research on threat detection through machine learning. This study proposes a method to extend and assess threat detection using machine learning for applications installed in the Android OS. The proposed scheme is *Malware Awareness (Level 1)* aimed at detecting malicious behavior for Android application, *Threat Awareness (Level 2)* for rating it, and *Decision-Making Awareness (Level 3)* for optimizing threat class.

The reasons for approaching from the viewpoint of SA are also related to CTI which is a recent issue. The availability of CTI is an essential element of threat assessment. The TARA developed by MITER also emphasizes the ability to identify countermeasures through threat assessment. In addition, cyber-SA framework research from the "Cybaware" project has resulted in an asset, configuration, impact, threat, and visualization as key areas of research [59]. In particular, the threat area has identified and evaluated the types of attackers (TTP, Tactics/Techniques/Procedures) and objectives and developed countermeasures as research results. Therefore, the proposed approach can contribute to threat detection,



(a) Number of threat class: 3

(b) Number of threat class: 5

(c) Number of threat class: 7

(d) Number of threat class: 10

Figure 6: Distribution of threats according to clustering of threat class.

production, measurement, and evaluation of CTI in the security field.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under Contract UD160066BD.

References

[1] J. Rivera, *Gartner Identifies The Top 10 Strategic Technology Trends for 2014*, Gartner, Inc, 2016.

[2] K. Panetta, *Inc. Retrieved at October*, Gartner, Inc., 2018.

[3] N. Hunn, "The Market for Hearable Devices 2016-2020," Technical Report, 2016.

[4] O. Gadyatskaya, F. Massacci, and Y. Zhauniarovich, "Security in the Firefox OS and Tizen mobile platforms," *The Computer Journal*, vol. 47, no. 6, pp. 57–63, 2014.

[5] P. R. Krishna, "Role of Internet of Things (IoT) in Entrepreneurship," 2017.

[6] W. C. Ashmore, "Impact of alleged Russian cyber attacks," *ARMY Command and General Staff Coll Fort Leavenworth Ks School of Advanced Military Studies*, 2019.

[7] P. B. Pathak and Y. M. Nanded, "A dangerous trend of cybercrime: ransomware growing challenge," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 5, pp. 371–373, 2016.

[8] M. Baezner, "Cyber disruption and cybercrime: Democratic Peoples Republic of Korea," *ETH Zurich*, vol. 9, 2018.

[9] A. Shabtai, L. Tenenboim-Chekina, D. Mimran, L. Rokach, B. Shapira, and Y. Elovici, "Mobile malware detection through analysis of deviations in application network behavior," *Computers & Security*, vol. 43, pp. 1–18, 2014.

[10] X. Yu, Z. Tian, J. Qiu, and F. Jiang, "A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5823439, 11 pages, 2018.

[11] F. Liu, C. Wang, A. Pico et al., "Measuring the insecurity of mobile deep links of Android," in *Proceedings of the26th USENIX Security Symposium (USENIX Security '17)*, pp. 953–969, 2017.

[12] N. Sombatruang, Y. Kadobayashi, M. A. Sasse, M. Baddeley, and D. Miyamoto, "The continued risks of unsecured public wi-fi and why users keep using it: evidence from japan," in *Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pp. 1–11, IEEE, Belfast, Northern Ireland, August 2018.

[13] M. La Polla, F. Martinelli, and D. Sgandurra, “A survey on security for mobile devices,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.

[14] N. Virvilis, N. Tsalis, A. Mylonas, and D. Gritzalis, “Mobile devices: A phisher's paradise,” in *Proceedings of the 2014 11th International Conference on Security and Cryptography (SECRYPT-2014)*, pp. 1–9, IEEE, Austria, August 2014.

[15] K. L. Chiew, K. S. C. Yong, and C. L. Tan, “A survey of phishing attacks: Their types, vectors and technical approaches,” *Expert Systems with Applications*, vol. 106, pp. 1–20, 2018.

[16] S. Na and T. Kwon, “RIK: A virtual keyboard resilient to spyware in smartphones,” in *Proceedings of the 2014 IEEE International Conference on Consumer Electronics, ICCE 2014*, pp. 21-22, January 2014.

[17] J. Cho, G. Cho, S. Hyun et al., “Open sesame! design and implementation of backdoor to secretly unlock android devices,” *Journal of Internet Services and Information Security (JISIS)*, vol. 7, no. 4, pp. 35–44, 2017.

[18] Joint Chief of Staffs, Joint publication 3-12 (R) Cyberspace Operations, pp. 2-4, 2013.

[19] D. Clark, “Characterizing Cyberspace: Past, Present And Future,” *MIT CSAIL*, vol. 1, pp. 1–4, 2010.

[20] R. Philip, *Enabling Distributed Security in Cyberspace*, Department of Homeland Security, 2011.

[21] M. Burhan, R. A. Rehman, B. Khan, and B.-S. Kim, “IoT elements, layered architectures and security issues: A comprehensive survey,” *Sensors*, vol. 18, no. 9, 2018.

[22] N. Koshizuka and K. Sakamura, “Ubiquitous ID: standards for ubiquitous computing and the internet of things,” *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 98–101, 2010.

[23] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, “Internet of Things: Security vulnerabilities and challenges,” in *Proceedings of the 20th IEEE Symposium on Computers and Communication, ISCC 2015*, pp. 180–187, July 2015.

[24] G. Aloi, G. Caliciuri, G. Fortino et al., “Enabling IoT interoperability through opportunistic smartphone-based mobile gateways,” *Journal of Network and Computer Applications*, vol. 81, pp. 74–84, 2017.

[25] A. D. Schmidt, H. G. Schmidt, L. Batyuk et al., “Smartphone malware evolution revisited: Android next target?” in *Proceedings of the 4th IEEE International Conference on Malicious and Unwanted Software*, pp. 1–7, 2009.

[26] M. Gigli and S. Koo, “Internet of Things: Services and Applications Categorization,” *Advances in Internet of Things*, vol. 1, no. 2, pp. 27–31, 2011.

[27] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, “Semantics for the internet of things: early progress and back to the future,” *International Journal on Semantic Web and Information Systems*, vol. 8, no. 1, pp. 1–21, 2012.

[28] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[29] T. Pahi, M. Leitner, and F. Skopik, “Analysis and assessment of situational awareness models for national cyber security centers,” in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy, ICISSP 2017*, pp. 334–345, February 2017.

[30] J. Okolica, J. T. McDonald, and G. L. Peterson, “Developing systems for cyber situational awareness,” *2nd Cyberspace Research Workshop*, pp. 46–56, 2009.

[31] G. P. Tadda and J. S. Salerno, “Overview of cyber situation awareness,” in *Advances in Information Security*, vol. 46, pp. 15–35, Springer, 2010.

[32] N. Evancich, Z. Lu, J. Li, Y. Cheng, J. Tuttle, and P. Xie, “Network-Wide Awareness,” in *Cyber Defense and Situational Awareness*, pp. 63–91, Springer, 2014.

[33] P. Barford, M. Dacier, T. G. Dietterich et al., “Cyber SA: Situational awareness for cyber defense,” in *Cyber Situational Awareness*, vol. 46, pp. 3–13, Springer, 2010.

[34] C. Atapour, I. Agrafiotis, and S. Creese, “Modeling Advanced Persistent Threats to enhance anomaly detection techniques,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 9, no. 4, pp. 71–102, 2018.

[35] J. Wynn, J. Whitmore, G. Upton et al., “Threat assessment remediation analysis (TARA): Methodology description version 1.0,” *Mitre Corp Bedford Ma*, Article ID MTR110176, 2011.

[36] J. Jones, “An introduction to factor analysis of information risk (fair),” *Norwich Journal of Information Assurance*, vol. 2, no. 1, 2006.

[37] I. Kotenko, I. Saenko, and A. Branitskiy, “Applying big data processing and machine learning methods for mobile internet of things security monitoring,” *Journal of Internet Services and Information Security (JISIS)*, vol. 8, no. 3, pp. 54–63, 2018.

[38] Z. Aung and W. Zaw, “Permission-based android malware detection,” *International Journal of Scientific &amp; Technology Research*, vol. 2, no. 3, pp. 228–234, 2013.

[39] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.-P. Wu, “DroidMat: android malware detection through manifest and API calls tracing,” in *Proceedings of the 7th Asia Joint Conference on Information Security (AsiaJCIS '12)*, pp. 62–69, August 2012.

[40] N. Khan, J. Abdullah, and A. S. Khan, “Towards vulnerability prevention model for web browser using interceptor approach,” in *Proceedings of the 9th International Conference on IT in Asia (CITA '15)*, pp. 1–5, August 2015.

[41] H. S. Ham, H. H. Kim, M. S. Kim et al., “Linear SVM-based android malware detection for reliable IoT services,” *Journal of Applied Mathematics*, vol. 2014, Article ID 594501, 10 pages, 2014.

[42] J. Sahs and L. Khan, “A machine learning approach to android malware detection,” in *Proceedings of the 2012 European Intelligence and Security Informatics Conference, EISIC '12*, pp. 141–147, August 2012.

[43] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, “Data mining methods for detection of new malicious executables,” in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 38–49, May 2001.

[44] K. Riad and L. Ke, “RoughDroid: Operative scheme for functional android malware detection,” *Security and Communication Networks*, vol. 2018, Article ID 8087303, 10 pages, 2018.

[45] I. Martín, J. A. Hernández, A. Muñoz et al., “Android malware characterization using metadata and machine learning techniques,” *Security and Communication Networks*, vol. 2018, Article ID 5749481, 11 pages, 2018.

[46] W. Niu, X. Zhang, G. Yang et al., “Identifying APT malware Domain based on mobile DNS logging,” *Mathematical Problems in Engineering*, vol. 2017, Article ID 4916953, 9 pages, 2017.

[47] L. Qiang, Y. Zeming, L. Baoxu, J. Zhengwei, and Y. Jian, “Framework of Cyber Attack Attribution Based on Threat Intelligence,” in *Interoperability, Safety and Security in IoT*, pp. 92–103, Springer, 2016.

[48] M. Park, J. Seo, J. Han et al., "Situational awareness framework for threat intelligence measurement of android malware," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 9, no. 3, pp. 25–38, 2018.

[49] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[50] J. Wu, "Cluster Analysis and K-means Clustering: an introduction," in *Advances in K-means Clustering*, pp. 1–16, Springer, 2012.

[51] D. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, pp. 827–832, 2015.

[52] I. D. Dinov, in *Expectation Maximization and Mixture Modeling Tutorial*, pp. 5–11, 2008.

[53] J.-W. Jang, H. Kang, J. Woo, A. Mohaisen, and H. K. Kim, "Andro-AutoPsy: anti-malware system based on similarity matching of malware and malware creator-centric information," *Digital Investigation*, vol. 14, pp. 17–35, 2015.

[54] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[55] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[56] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The Journal of Educational Research*, vol. 96, no. 1, pp. 3–14, 2002.

[57] D. M. Powers, *Evaluation: from Precision, Recall And F-Measure to ROC, Informedness, Markedness and Correlation*, 2011.

[58] J. Meakins, "A zero-sum game: the zero-day market in 2018," *Journal of Cyber Policy*, pp. 1–12, 2018.

[59] R. A. Kemmerer, *Cybaware: A Cyber Awareness Framework for Attack Analysis, Prediction, and Visualization*, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 4064201, 12 pages
https://doi.org/10.1155/2019/4064201

Research Article

Radio Environment Map Construction by Kriging Algorithm Based on Mobile Crowd Sensing

Zhifeng Han, Jianxin Liao, Qi Qi, Haifeng Sun, and Jingyu Wang

Institute of Network Technology, Beijing University of Posts and Telecommunications, 100088, China

Correspondence should be addressed to Zhifeng Han; zhifeng.han@gmail.com

Received 25 October 2018; Revised 28 December 2018; Accepted 13 January 2019; Published 3 February 2019

Academic Editor: Ilsun You

In the IoT era, 5G will enable various IoT services such as broadband access everywhere, high user and devices mobility, and connectivity of massive number of devices. Radio environment map (REM) can be applied to improve the utilization of radio resources for the access control of IoT devices by allocating them reasonable wireless spectrum resources. However, the primary problem of constructing REM is how to collect the large scale of data. Mobile crowd sensing (MCS), leveraging the smart devices carried by ordinary people to collect information, is an effective solution for collecting the radio environment information for building the REM. In this paper, we build a REM collecting prototype system based on MCS to collect the data required by the radio environment information. However, limited by the budget of the platform, it is hard to recruit enough participants to join the sensing task to collect the radio environment information. This will make the radio environment information of the sensing area incomplete, which cannot describe the radio information accuracy. Considering that the Kriging algorithm has been widely used in geostatistics principle for spatial interpolation for Kriging giving the best unbiased estimate with minimized variance, we utilize the Kriging interpolation algorithm to infer complete radio environment information from collected sample radio environment information data. The interpolation performance is analyzed based on the collected sample radio environment information data. We demonstrate experiments to analyze the Kriging interpolation algorithm interpolation results and error and compared them with the nearest neighbor (NN) and the inverse distance weighting (IDW) interpolation algorithms. Experiment results show that the Kriging algorithm can be applied to infer radio environment information data based on the collected sample data and the Kriging interpolation has the least interpolation error.

1. Introduction

In the IoT networks, 5G technology is characterized by higher bit rates with more than 10 Gigabits per second as well as by more capacity and very low latency, and it will leverage novel technological concepts to meet the "anywhere and anytime" requirements of IoT devices. With the rapid development of IoT devices amount, the demand for wireless spectrum resources is increasing. In order to dynamically plan the spectrum resources to improve the utilization of radio resources to provide well control of IoT devices' access control, we can build radio environment map (REM) to collect and understand the radio information. REM can offer multidomain environmental information, such as geographical features, available services, spectral regulations, locations and activities of radios, relevant policies, and experiences [1].

However, the primary problem of constructing REM is how to collect large scale of data. Currently, most of the REMs are aimed at small scale and applied to specific applications. And the universal methods to build a REM are by deploying sensors in a certain environment to collect the sensing data. However, the REM is applied to dozens of different kinds of networks and applications, which makes the networks and applications have to collect data separately [2, 3]. Besides, the same data can hardly be shared and reused among different applications, resulting in duplication of data collection and a waste of resources. Therefore, it is of great significance to construct a large scale and universal REM, which can integrate data sources of radio environment and avoid the cost of the reconstructing database [4]. Mobile crowd sensing (MCS) is an effective solution to solve this problem, which is a novel emerging paradigm that leverages the smart devices carried by ordinary people to collect information and has facilitated many sensing applications, such as environment monitoring, traffic detection, social interaction, and public information sharing [5]. MCS can be applied to collect radio environment information in the sensing area. In order to characterize environmental information comprehensively, recruiting adequate ordinary users with smart devices to participate in radio environment information collection is needed. Compared with the traditional data collection technologies, MCS collects the environment information by built-in sensing modules in the mobile terminals, and it has the properties of mobility, the ubiquity of nodes, the powerful storing, and computing ability [6, 7].

Wireless network signals are all electromagnetic waves, whose transmission and attenuation are complex process. Therefore, in this paper, we only analyze the transmission and attenuation processes of electromagnetic waves in space entropy in ideal conditions. Under ideal conditions, the propagation process is free from any obstruction and without any multipath propagation. Then the propagation model of space electromagnetic waves is a free space propagation model. According to the pattern of wireless electromagnetic wave transmission in free space, spatial interpolation algorithm can be applied to restore the uncollected radio environment information data of the sensing area. The Kriging interpolation algorithm has been widely used in geostatistics principle for spatial interpolation but is not broadly used in wireless network area. Kriging spatial interpolation algorithm estimates unknown point data and not only considers the relative positions of estimated points and known sample points, but also considers the relative positional relationship between all sample points. In this paper, we proposed to use Kriging interpolation algorithm to infer the uncollected radio environment information based on the collected sample data.

In this paper, we proposed to apply the MCS to collect the radio information. Furthermore, to address the problem of the incomplete radio environment information caused by the inadequate sensing data, we proposed to apply the Kriging interpolation algorithm to infer the uncollected radio environment information with the collected sensing data. Our contributions are as follows:

(i) We propose a REM prototype system based on MCS, where the ubiquitous, massive, and high dimension REM-related data can be sensed and collected by the terminals carried by mobile users.

(ii) We propose to apply Kriging interpolation algorithm to infer the uncollected radio environment information caused by the target area being not covered by the participants.

(iii) We set up experiments to collect the sample of the radio environment information and infer the missing radio information of the target area. The results show that the Kriging interpolation algorithm can infer the missing radio information and has the least interpolation error.

The rest of the paper is organized as follows. In Section 2, the related works are introduced. Section 3 outlines the architecture of the REM based on MCS. In Section 4, the Kriging interpolation algorithm is introduced. The simulation results are illustrated in Section 5. Section 6 presents the conclusion.

2. Related Works

Building REM needs a large number of sensors and kinds of radio environment information, which is a great challenge. At present, data collection methods for REM can mainly be categorized into three types. First is integrating or accessing the related information directly from existing databases, estimating radio propagation characteristics by software tools, and leveraging cognitive radios devices or networks to sense data. Gathering data from the existing database is a relatively convenient way, while the data updating time depends on the updating period of the underlying database. Moreover, the historical information is not stored in the underlying database. Riihijärvi et al. take vantage external datasets to build REM, but the update cycle of the external datasets is very long which makes datasets unable to meet the real-time requirement of REM [8]. Constructing REM in this way makes it difficult to satisfy the upper-layer applications with the requirement for real-time and historical information. Second, the way to characterize and estimate the properties of radio transmission based on software is to calculate the signal attenuation by modeling so that we can better plan the radio environment [9, 10]. The model in [11] clearly gives a solution to the signal diffraction problem caused by the occlusion, but this requires an accurate vector model of all three-dimensional structures, with limited data and resolution in most experimental environments. It cannot be applied to applications that require high accuracy. The above-mentioned estimation method usually provides limited data, bad accuracy of the data. Third, the method based on wireless device or external network mainly uses the information sensing ability of heterogeneous spectrum sensor network to collect data [12, 13]. In terms of data collection, MCS refers to the sensing paradigm in which mobile users with sensing and computing devices are tasked to collect and contribute data in order to enable various applications [14]. It combines people-centric sensing and crowdsourcing so that a great number of ordinary users with smart devices can cooperate with each other to form a sensing network and deliver the sensing tasks [5]. Then participants can upload the sensing data to the MCS platform. The development of MCS has resulted in various novel sensing applications. Some typical examples include the air quality inspection application Common Sense for air quality monitoring [5] by the University of California Berkeley, the Creek Watch application to evaluate city water resources [15] by IBM, and the Nericell system [16] by Microsoft to monitor road and traffic condition implemented by piggybacking on smartphones that users carry with them in normal course. MCS has attracted much attention from researchers due to advantages such as ubiquitous sensor nodes, good participant mobility, low maintenance cost, and rich sensing data types. Hence, MCS can be applied to collect the radio environment information.

MCS can be applied to collecting large scale data due to the properties of mobility, the ubiquity of sensing nodes. However, limited by the budget, there are no enough participants recruited to join the data collection, which makes the radio environment information incomplete. To build the REM, complete radio information is needed. A lot of works have been done about inferring the missing data according to the sample data in many research fields. Talvitie et al. investigate spatial interpolation and extrapolation algorithms for construction of fingerprint databases [17]. Lacking knowledge about the beacon locations, measurement at an unknown point is interpolated based on actual measurements in the surrounding. There are several interpolation algorithms considered in [17], which include linear interpolation based on Delaunay triangulation, the nearest neighbor (NN), and the inverse distance weighting (IDW) to name a few. The results show that location accuracy is enhanced by utilizing constructed databases comparing to the incomplete database. Grimoud et al. use an iterative process to obtain the REM based on Kriging interpolation to reduce the measurement data required [18]. Umbert et al. apply Kriging and a modified version of the Inverse Distance Weighted (IDW) algorithm to build a REM of an outdoor TV spectrum resources [19]. Hence, there is a spatiotemporal correlation between radio environment data, and the Kriging interpolation algorithm can be applied to infer the missing radio environment information data. Here, we use Kriging interpolation algorithm to infer the missing radio environment information data according to the collected sample data.

3. The REM Based on MCS Architecture

In this section we will introduce the REM based on MCS. First, we present the system architecture and discuss the system components of the radio environment information data collection platform. Second, we introduce the data collection process used to collect the radio environment information data.

3.1. Radio Environment Information Collection System Architecture

3.1.1. Radio Environment Information Collection Platform. Figure 1 shows the overview of our system based on MCS. As shown in Figure 1, from bottom to upper layer, the system includes data sensing layer, data collection layer, data processing layer, data analysis layer and visualization layer. In the data sensing layer, a large number of mobile terminals constitute the mobile crowd sensing network, and they play the role of data sensing by running our data collecting APP named wireless detect. The mobile terminals upload the sensing data to our cloud servers via Wi-Fi/3G/4G networks. The data collection layer is mainly responsible for receiving data, node selection, task allocation, and making incentive mechanism to recruit enough interested nodes to participate in the sensing tasks. The data preprocessing includes arranging the data format and data fusion. The data analysis layer is responsible for the statistical analysis and calculation of the radio environment relevant parameters. At last, the visualization layer shows the REM relating results in the forms of the field strength map, heat map, and some other maps.

Our proposed architecture involves various functional blocks, communicating via well-specified interfaces. To establish a complete radio environment map, the fundamental problem is the collection of a large number of data with complex types and data processing and visualization. Our system consists of five different function modules: data sensing, data collection, data processing, data analysis, and visualization; each of them has its own function.

Data sensing module is operated by the MCS network, which is organized by mobile terminals carried by mobile users. When a mobile user receives a data sensing task, it will determine whether the user is involved in the task. If so, it will collect the required data by the sensing module embedded in the terminal. Moreover, it also uploads data to the web server by different types of network accessing technologies like Wi-Fi/3G/4G. Our system includes perception of user-uploaded data and calls, mobile phone map API, real-time construction of heat map, and signal strength map. Users can use wireless detection real-time view of the environment in which the radio spectrum resources are used.

Data collection mainly includes area partition, incentive mechanism, nodes selection, task distribution, data storage, and data distribution. The area partition is designed to identify whether a sensing task refers to a geographical location or is based on some social relationships. In our system, we divided it into regional division and business division. The incentive mechanism is used to reduce the cost of the platform as well as attracting enough sensing users. Furthermore, node selection mechanism needs to select the appropriate node for the data sensing and also needs to assign the sensing nodes to the corresponding sensing tasks if there is more than one task.

Data processing module mainly includes two modules: data preprocessing (filtering and cleaning) and data fusion, which is implemented by the MapReduce workflow. The data processing flow is as follows. Firstly, the Avro in the data fusion module compresses various types of formats of the data and merges massive small files into large files to improve the efficiency of MapReduce. Secondly, as the raw data is varying in data types, the data cleaning and filtering can play an important role to remove the noise and interference such as error data. Thirdly, these data are processed by sever cluster, and the processing results are stored in data center.

Data analysis is responsible for the statistical analysis and calculation after the data preprocessing. In order to exhibit the radio environment on the map, it needs to perform analysis and calculation to get the related parameters such as the channel occupation, frequency band occupancy, and background noise intensity.

The visualization module is responsible for the REM-related data parameters exhibition. We designed the visualization for the REM properties. The system can show the Wi-Fi signal coverage, cellular signal coverage heat map, and Wi-Fi channel occupation ratio map. The visual REM

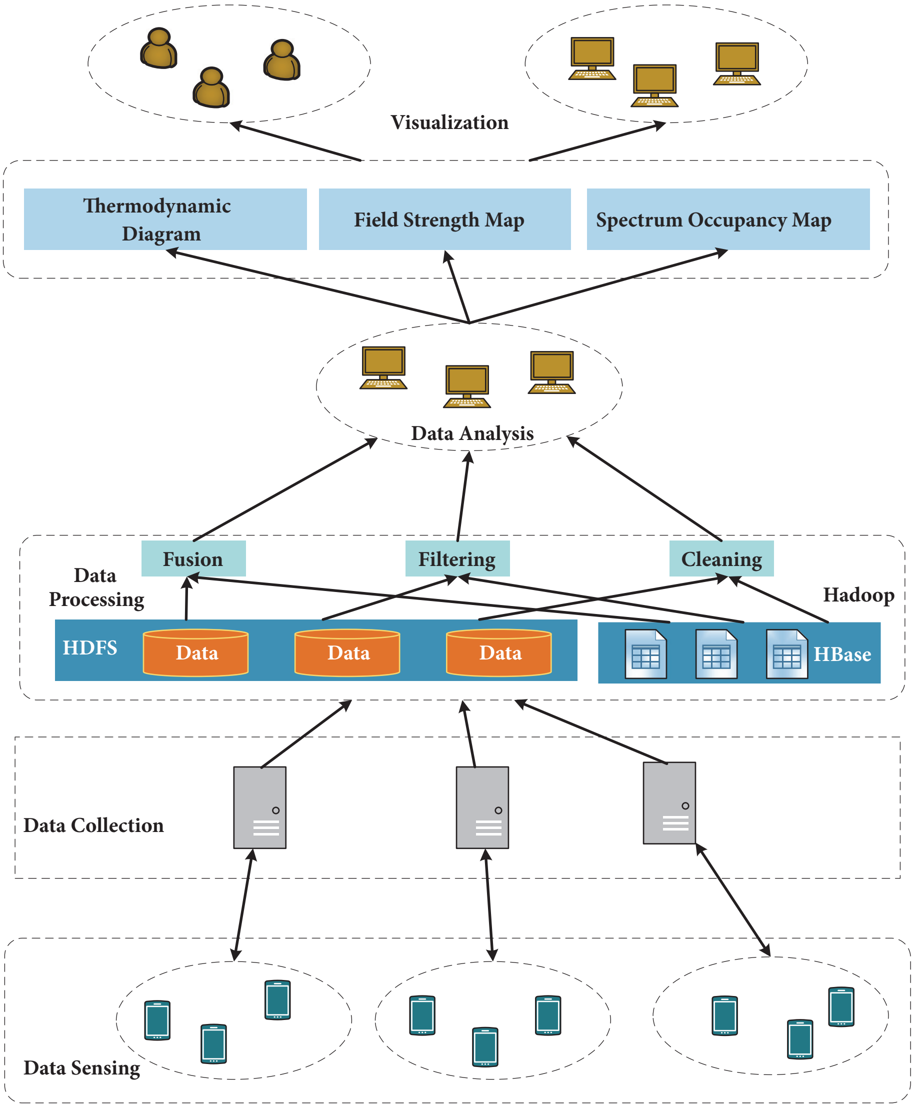


Figure 1: Radio information collection architecture based on MCS.

makes it easy to identify the radio environment of the target area.

3.2. Radio Environment Information Collection Process. In [20], the author proposed 4W1H model in mobile sensing and divided the MCS life cycle into four phases, which is shown in Figure 2: task creation, task assignment, individual task execution, and crowd data integration according to the MCS life cycle. Next, we will discuss the following key design issues: REM task creation, REM task assignment, participants recruiting, and participants' selection.

The task creation specifies the sensing timing and coverage area for the REM. In our system, the web server releases the sensing tasks to the users who are interested in the data collection task. REM supports long spatiotemporal information for the upper-layer applications, so the sensing time is continuous.

In our REM task assignment stage, the system is responsible for recruiting and selecting participants for the MCS task. Correspondingly, this stage includes participants recruiting, participants' selection, and incentive mechanism. We choose the well suited participants to join the sensing task to collect the radio environment information, and reward them for the high quality sensing data. The purpose of participants recruiting is to encourage enough people to join the sensing task and get more radio environment data. However, limited by the budget of the platform or the human mobility, only part of the participants can join the radio environment information sensing task. Then the radio environment information data is incomplete; we will talk about the solution later.

In sensing task execution, participants conduct sensing tasks and upload the sensed data to the MCS platform. The participants receive the sensing tasks and then collect the radio environment data. The selected participants are

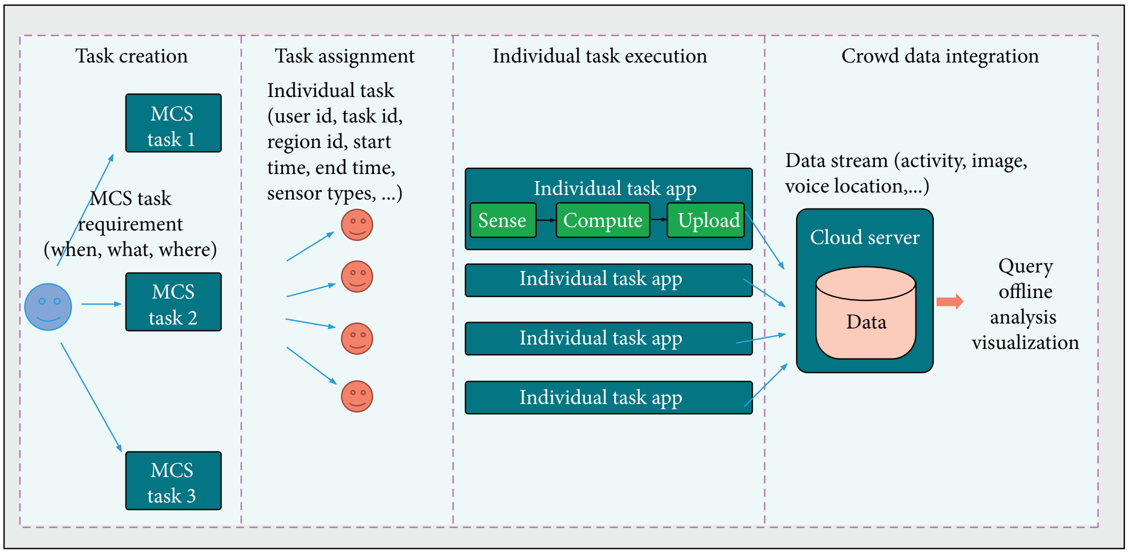

Task creation
Task assignment
Individual task execution
Crowd data integration
MCS task 1
MCS task 2
MCS task 3
MCS task requirement (when, what, where)
Individual task (user id, task id, region id, start time, end time, sensor types, ...)
Individual task app
Sense
Compute
Upload
Individual task app
Individual task app
Individual task app
Data stream (activity, image, voice location,...)
Cloud server
Data
Query offline analysis visualization

Figure 2: The four-stage life cycle of the mobile crowd sensing process [20].

distributed in the target places collecting data. After radio environment data collection the participants upload the data to the MCS platform server by cellular networks (3G/4G) or WLAN.

During the data integration, the main issue is to achieve the MCS task that is to process and analyze the raw data received from mobile terminals and visualize the required results eventually.

4. Missing Data Inference by Kriging

4.1. Related Definition. In this section we will introduce how to infer the missing radio environment information data by Kriging interpolation algorithm. The whole process is shown in Figure 3, which can be divided into two steps. First, we analyze the distribution of sample points distributed in the sensing area and propose a variogram model to reflect the spatial structure characteristics and distribution characteristics of the variables. Second, we use Kriging algorithm to calculate the missing data according to the collected sample radio environment information data. Third, some basic concepts and related definitions are introduced, and all the parameters are listed in Table 1.

Definition 1 (variogram). In spatial statistics the theoretical variogram $2\gamma(s_1, s_2)$ is a function describing the degree of spatial dependence of a spatial random field or stochastic process $Z(s)$. Given an area of interest $G \subset R^n$, the mean of RSS value at a location x_i is considered as a random variable (RV) $\overline{Z}_i$. Then, the mean of RSS values over the area can be represented by a random field (RF), which is a collection of spatial RVs, $\{\overline{Z}_i \mid X_i \in G\}$.

Definition 2 (stationary process). Formally, let $\{X_t\}$ be a stochastic process and let $F_X\{x_{t_1+\tau}, \cdots, x_{t_k+\tau}\}$ represent the cumulative distribution function of the unconditional (i.e., with no reference to any particular starting value) joint distribution of $\{X_t\}$ at times $t_1 + \tau, \cdots, t_k + \tau$. Then, $\{X_t\}$ is said to be strictly (or strongly) stationary if, for all k, for all τ, and for all $t_1, \cdots t_k$, $F_X(x_{t_1+\tau}, \cdots, x_{t_k+\tau}) = F_X(x_{t_1}, \cdots x_{t_{kss}})$.

Table 1: Description of parameters in the equations.

Symbols	Descriptions
$\overline{Z}_i$	Random variable
$\delta_{i,j}$	Difference between two neighbor points
$E[\delta_{i,j}]$	Mathematical expectation
$\gamma(h)$	Variogram function
$N(h)$	Number of pairs of observations
h	The separation distance
$\gamma_{i,j}$	Variogram value
G	A set representing the area of interest
α, β	Fitting parameters with a strict constraint
z_j^i	RSS value received from the jth beacon

4.2. Problem Formulation

(1) Analysis of the distribution of sample points:

Given the sample radio environment information data D_s of the area $G \subset R^n$, we need to use the variogram function to analyze the distribution of sample data in the sensing area. If the sample data only depends on the distance h between the sample data points, we can use Kriging interpolation to infer more data.

(2) Kriging interpolation process:

Given the sample radio environment information data D_s of the area $G \subset R^n$, we need to infer the complete radio environment information data D_c to build the REM.

4.3. The Variogram. First, some basic concepts and definitions of the Kriging interpolation algorithm are introduced as follows:

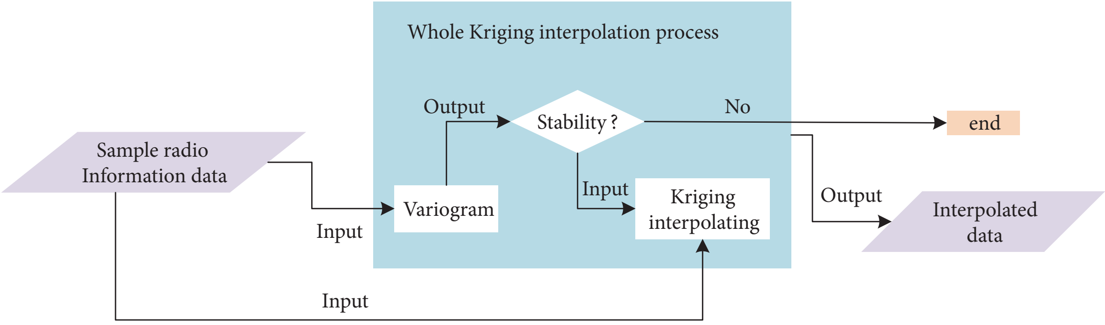


Figure 3: The whole process of inferring more radio information data by Kriging interpolation algorithm.

The variogram is often applied in the statistical process of geolocation-related information. variogram can describe the structural change and distribution of variables in the geospatial space. Assume that the value of sample data point x in the sensing area is $Z(x)$ and the sample data value of point $x + h$ is $Z(x + h)$. Then half of the variance of the difference between the values at the two points is defined as the variation of $Z(x)$ at position x. The function can be expressed as

$$\begin{aligned} r(x,h) &= \frac{1}{2}\operatorname{var}\left\{[Z(x) - Z(x+h)]^2\right\} \\ &= \frac{1}{2}\Big(E[Z(x) - Z(x+h)]^2 \\ &\quad - \{E[Z(x) - Z(x+h)]\}^2\Big) \end{aligned} \tag{1}$$

where $r(x, h)$ is the variogram and $Z(x)$ and $Z(x+h)$ are sample attribute values of the variables at points x and $x + h$ in the target area, respectively. h is the distance between x and $x + h$, and $E\{[Z(x) - Z(x + h)]^2\}$ is the mathematical expectation. In the variogram function, when the increment $Z(x)$ of the variable $[Z(x)-Z(x+h)]$ of the target area satisfies the following two conditions, it is said that $Z(x)$ satisfies the second-order stationarity.

(i) First, the sample radio environment information data have to satisfy the rules, which are listed as follows:

$$E[Z(x)] = E[Z(x+h)] = m, \quad \forall x, \forall h \tag{2}$$

means the target regionalization variable does not have obvious characteristics in terms of space and fluctuates around m.

(ii) Second, in the entire target area, the covariance function of $Z(x)$ exists and is stable; namely,

$$\begin{aligned} \operatorname{cov}[Z(x), Z(x+h)] &= E[Z(x)Z(x+h)] \\ &\quad - E[Z(x)]E[Z(x+h)] \\ &= E[Z(x)Z(x+h)] - m^2 \\ &= C(h). \end{aligned} \tag{3}$$

The attribute value $Z(x + h)$ in the target area has no relationship with the position point x and is only related to h, which means the value is relative to the relative position and does not depend on the absolute position. According to the above two conditions, it can be concluded that the target regionalization variable is strictly second-order stationary in the target area. However, since it is difficult to satisfy a strict second-order stationary state in real life, the condition that satisfies the strict second-order stationary state is weakened to obtain an intrinsic assumption, also called an intrinsic assumption. Similarly, when the increment $[Z(x) - Z(x + h)]$ of the target regionalization variable $Z(x)$ satisfies the following two conditions, it is said to satisfy the intrinsic assumption:

(i) First, the sample radio environment information data have to satisfy the rules, which are listed as follows.

$$E[Z(x) - Z(x+h)] = 0 \tag{4}$$

(ii) Second, the variance function of the target regionalized increment $[Z(x) - Z(x + h)]$ exists and is stable; the variance function of the increment does not depend on x, then

$$\begin{aligned} &\operatorname{var}[Z(x) - Z(x+h)] \\ &\quad = E[Z(x) - Z(x+h)]^2 - \{E[Z(x) - Z(x+h)]\}^2 \\ &\quad = E[Z(x) - Z(x+h)]^2, \quad \forall x, \forall h. \end{aligned} \tag{5}$$

When the variation of the target area satisfies the weak second-order stationary or intrinsic assumption, due to $E[Z(x) - Z(x + h)] = 0$, the half-difference function can be expressed as follows.

$$(r,h) = \frac{1}{2}E[Z(x) - Z(x+h)]^2 \tag{6}$$

At this time, the increment $[Z(x) - Z(x + h)]$ of $Z(x)$ is only related to the distance between two points. The above variogram function is a theoretical variogram function. In actual operation, multiple sample data needs to be divided into multiple pairs for calculation, like $\{Z(x_i), Z(x_i + h)\}(i = 1, 2, \ldots, N(h))$, where $N(h)$ is the number of pairs of points in the sample data divided by h. The extremes of the variogram can be calculated in the following way.

$$\gamma^{*}(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} \left[Z(x_i) - Z(x_i + h)\right]^2 \tag{7}$$

In (7), h is the distance between the sample point and the point to be estimated, and $N(h)$ is the number of samples used to calculate the variogram of the sample between $(x_i, x_i + h)$. After the above steps, the analysis of the distribution characteristics of sample points in the target area has been completed. However, in order to estimate the unknown value of the target area variable, the fitting of the convenience function point of the actual sample is also called the theoretical variogram model. The theoretical model of the variogram is to abstract the experimental variogram and then use it to calculate the Kriging interpolation.

The empirical variogram contains values at a limited number of h. To estimate the measurements at unknown locations, access to the value of h between the scattered points in the empirical variogram is required. Hence, a mathematical model is selected to be fitted in the empirical variogram. This model is frequently chosen from spherical model, exponential model, Gaussian model, power model, and linear model. We choose the sample radio environment data and input them to the Matlab. Then we can use the fitting function to find a mathematical expression.

4.4. Kriging Interpolating. Once the variogram is obtained, values at unknown locations can be estimated based on known data points. Mathematically, this problem can be regarded as a spatial interpolation problem. Assuming that the target area to be studied is A, the variable in the target area is $\{Z(x) \in A\}$, where x represents a position in the target area. The sample value of $z(x)$ in the target area $x_i (i = 1, 2, \cdots, n)$ is $z(x_i)(i = 1, 2, \cdots, n)$. Then the value $z(x_0)$ at the point x_0 to be estimated is the weighted sum of the known n point sampling values:

$$z(x_0) = \sum_{i=1}^{n} \lambda_i z(x_i) \tag{8}$$

where $\lambda_i (i = 1, 2, \cdots, n)$ is the weight coefficient of the known sample point. Due to the fact that $z(x)$ satisfies the second-order stationary assumption when analyzing the distribution of sample points in the target area, then

(i) there is a mathematical expectation for variable $z(x)$, and the expected value is a constant, $E[z(x)] = m$;

(ii) there is a covariance function for the variable A; that is, the value of the point to be estimated in the target area is only related to the distance B between the positions of the known sample points; then,

$$\begin{aligned} \operatorname{cov}\{z(x), z(x+h)\} &= E[z(x) \cdot z(x+h)] - m^2 \\ &= C(h). \end{aligned} \tag{9}$$

According to the unbiased requirements of the interpolation available,

$$E[z^{*}(x_0)] = E[z(x_0)]. \tag{10}$$

Then,

$$\sum_{i=1}^{n} \lambda_i = 1. \tag{11}$$

Then under the condition that $z(x)$ is second-order stationary, the calculation process of the estimated variance can be performed by the following method.

$$\begin{aligned} \sigma_E^2 &= E[z^{*}(x_0) - z(x_0)]^2 - \{E[z^{*}(x_0) - z(x_0)]\}^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j C_{i,j} - 2\sum_{i=1}^{n} \lambda_i C_{i,0} + C_{0,0} \end{aligned} \tag{12}$$

In order to minimize the variance of unbiased estimates, $\min\{\operatorname{var}[z^{*}(x_0) - z(x_0)] - 2\mu \sum_{i=1}^{n} (\lambda_i - 1)\}$, and then we can get the equations for the weighting coefficients λ_i in (8).

$$\begin{gathered} \sum_{i=1}^{n} \lambda_i \operatorname{cov}(x_i, x_j) + \mu = \operatorname{cov}(x_0, x_i) \\ \sum_{i=1}^{n} \lambda_i = 1, \quad i = 1, 2, \cdots, n \end{gathered} \tag{13}$$

The equations are the Kriging equations. In addition, (8) is written in matrix form:

$$[K][\lambda] = [M] \tag{14}$$

where

$$\begin{gathered} [\lambda] = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ -\mu \end{bmatrix} \\ [M] = \begin{bmatrix} C_{01} \\ C_{02} \\ \vdots \\ C_{0n} \\ 1 \end{bmatrix} \\ [K] = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} & 1 \\ C_{21} & C_{22} & \cdots & C_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}. \end{gathered} \tag{15}$$

After solving the weight coefficients λ of the above equations, you can use (8) to calculate and calculate the predicted values with the valuation points. Since the Kriging interpolation algorithm has a minimum estimation error based on known samples and is considered according to the distribution of the attribute values of the target region, the data of more known sample points within the target region can be used for estimation, and the estimated values are closer to the true value.

We apply the Kriging interpolation algorithm to infer more radio environment information data according to the sample collected information. We use the sample data to estimate the value of sensing area. Our Kriging weights are derived through minimizing the estimator error variance; that is,

$$\min_{\lambda_i \in R} \operatorname{var}\left(\overline{z}_u^* - \overline{z}_u\right) \tag{16}$$

under the unbiasedness constraint, given by the following.

$$E\left[\overline{z}_u^* - \overline{z}_u\right] = 0 \tag{17}$$

The mathematical expectation of the sample radio environment information is zero. Assuming the intrinsic stationarity and utilizing Lagrange multiplier optimization algorithm to minimize the estimator error variance (16) under the unbiasedness constraint (17), the Kriging weights λ_i in (8) can be calculated as

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \\ L \end{pmatrix} = \begin{pmatrix} \gamma_{1,1} & \cdots & \gamma_{1,k} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma_{k,1} & \cdots & \gamma_{k,k} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} \gamma_{1,u} \\ \vdots \\ \gamma_{k,u} \\ 1 \end{pmatrix} \tag{18}$$

where $\gamma_{i,j}$ is the radio environment information variogram value between the ith and jth neighbor data points, $\gamma_{i,u}$ is the radio environment variogram value between the ith neighbor data point and the interpolation point.

5. Simulation Evaluation

In this section, we show the prototype REM system based on MCS and demonstrate the simulation results of the inference of the missing radio environment information data according to the sample data.

5.1. Implementation of the Prototype System. As is shown in Figure 4, the radio environment information is collected and displayed in the web portal of the platform. As we can see several Wi-Fi properties can be seen on the banner. The properties collected by participants are as follows: SSID, BSSID, frequency, and the Wi-Fi signal strength level. The information belongs to the Wi-Fi signal sources sensed by the nearby participants. The density of the red nodes represents the density of the Wi-Fi signal sources. As we can see in Figure 8 the density of the Wi-Fi signal source is not uniform. This result is as expected.

Table 2: Experimental equipment parameter configuration.

AP	MAC	channel	Transmission power (dBm)
AP1	48:7A:DA:B9:21:A0	11	20dBm
AP2	48:7A:DA:B9:12:60	1	20dBm
AP3	50:DA:00:9D:B3:E0	6	20dBm

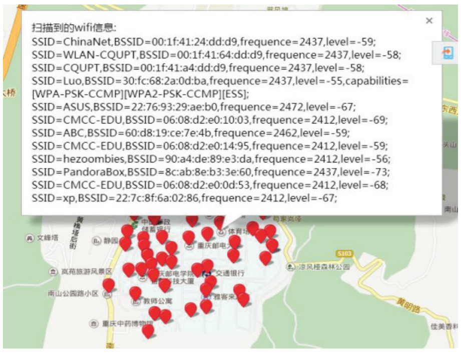
5.2. Interpolation Performance Evaluation. In this section, experimental settings are described in detail. We will introduce the experiment environment first, and the baseline methods used in the experiments are presented. The experimental data is also introduced, and experimental settings and evaluation metrics are also proposed to evaluate the performance of our method.

5.2.1. Experimental Settings. The experimental area is 150m× 150m as the sensing area to collect the radio environment information and infer the missing radio environment information data. In order to simulate the wireless network environment of the target area, three wireless network access points (APs) and radio network controllers (ACs) are deployed to form a WLAN in the target area to cover the target sensing area. The AP and AC are H3C WA4320-ACN and H3C WX3010E. In our WLAN network, the frequency band of the electromagnetic wave transmitted by the AP is 2.4 GHz band. In order to better cover the target area and reduce mutual interference between APs, the three APs use the 1, 6, and 11 channels of the 2.4G band respectively, and the transmission power of AP electromagnetic waves is 20 dBm. All parameters are listed in Table 2.

In order to simulate a small number of users collecting radio environment information in the target area, we mesh the target area. We divide the target area into subareas, each of which has a size of $5m \times 5m$, and then the participants collect the WLAN signal in different subareas using mobile devices to collect Received Signal Strength Indication (RSSI), as shown in Figure 7. The device used by the participants to collect the RSSI value is a Lenovo smart phone (Lenovo A3910e70), and the Wi-Fi analyzer is used to obtain the received signal strength value of the wireless network in the subarea. In order to prove that using the Kriging spatial interpolation algorithm to infer the wireless network environment data in our REM platform has higher accuracy, control experiments are set up. In the control experiments, the missing wireless network environment in the target area is restored by Nearest Neighbor (NN) and Inverse Distance Weighting (IDW) according to the sample data of the wireless network environment data collected by the participants.

5.2.2. Baseline Methods. To verify the high accuracy of the Kriging spatial interpolation algorithm, we used the nearest neighbor (NN) interpolation algorithm, inverse distance weighted (IDW) interpolation algorithm, and Kriging interpolation algorithm to predict the restoration goal under the same data volume of sample data.

(a) NN [21, 22]: Nearest neighbor interpolation is a simple method of multivariate interpolation in one or more



```
扫描到的wifi信息:
SSID=ChinaNet,BSSID=00:1f:41:24:dd:d9,frequence=2437,level=-59;
SSID=WLAN-CQUPT,BSSID=00:1f:41:64:dd:d9,frequence=2437,level=-58;
SSID=CQUPT,BSSID=00:1f:41:a4:dd:d9,frequence=2437,level=-58;
SSID=Luo,BSSID=30:fc:68:2a:0d:ba,frequence=2437,level=-55,capabilities=
[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS];
SSID=ASUS,BSSID=22:76:93:29:ae:b0,frequence=2472,level=-67;
SSID=CMCC-EDU,BSSID=06:08:d2:e0:10:03,frequence=2412,level=-69;
SSID=ABC,BSSID=60:d8:19:ce:7e:4b,frequence=2462,level=-59;
SSID=CMCC-EDU,BSSID=06:08:d2:e0:14:95,frequence=2412,level=-59;
SSID=hezoombies,BSSID=90:a4:de:89:e3:da,frequence=2412,level=-56;
SSID=PandoraBox,BSSID=8c:ab:8e:b3:3e:60,frequence=2437,level=-73;
SSID=CMCC-EDU,BSSID=06:08:d2:e0:0d:53,frequence=2412,level=-68;
SSID=xp,BSSID=22:7c:8f:6a:02:86,frequence=2412,level=-67;
```

Figure 4: The web portal of REM based on MCS.

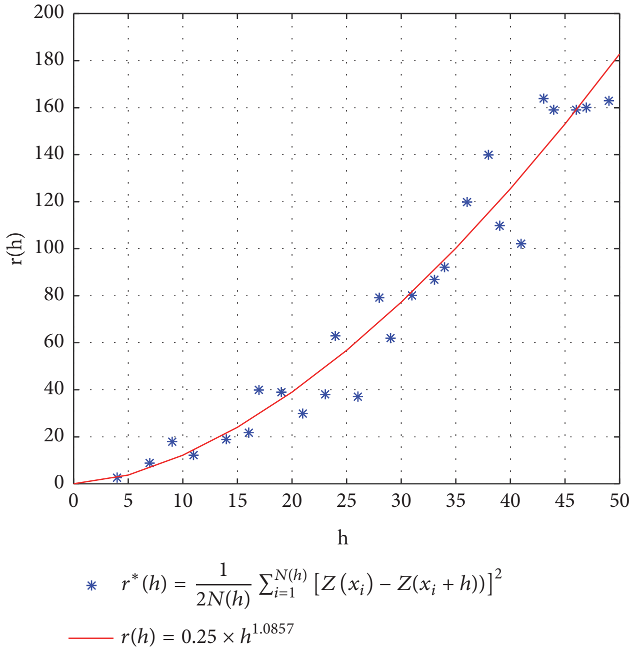


Figure 5: Variation function discrete data points and fitting curves.

dimensions. For a given set of points in space, a Voronoi diagram is a decomposition of space into cells, one for each given point, so that anywhere in space, the closest given point is inside the cell. This is equivalent to nearest neighbor interpolation, by assigning the function value at the given point to all the points inside the cell.

(b) IDW [23]: Inverse distance weighting (IDW) is a type of deterministic method for multivariate interpolation with a known scattered set of points. The assigned values to unknown points are calculated with a weighted average of the values available at the known points.

5.2.3. Experimental Data. In order to obtain the discrete data points calculated from the experimental variogram based on the WLAN sample data collected in the target area, a theoretical variogram model and a power function model were selected according to the distribution of the discrete data points. To verify the effect of different numbers of sample data on inference of the radio environment data of the entire target area, the target area is divided into 315 subareas, and a scale factor of the number of sample data and total data is set. During the experiment, the WLAN emission electromagnetic wave signal propagates in the free space and causes attenuation. Therefore, the smaller the distance from the electromagnetic wave transmission position, the greater the RSSI value that can be obtained. We choose $\rho = 0.1$ and $\rho = 0.3$ to indicate the percentage (10% and 30%) of subareas being covered by the participants of the target area, respectively; by drawing the target area WLAN signal RSSI heat map, we can compare the accuracy of our algorithm under different coverage situations. According to the sample data collected from the sample, the heat map of the RSSI is plotted using the Kriging spatial interpolation algorithm.

Figure 5 presents the empirical variogram and fitting result of a beacon in radio environment information. We input the sample information data to Matlab. The fitted curve demonstrates the spatial correlation model of data and is used to estimate the information data at a sensing location. As shown, the value of empirical variogram, which is the scatter plot in Figure 7, increases with h. It infers that there is an obvious trend (general spatial variation of the mean value) of RSS distribution in the area. Compared with more widely used fitting functions, e.g., the spherical and exponential function, it is suggested that a power model is selected; that is,

$$\gamma(h) = \begin{cases} 0, & h = 0 \\ \alpha h^{\beta}, & h \geq 0 \end{cases} \tag{19}$$

where α and β are the fitting parameters with a strict constraint that $0 < \beta < 2$.

As indicated in the figure, the power function is well fitted, so we choose the power function as the variogram. It can be obtained by fitting with Matlab fitting function, as is shown in Figure 5. After the Matlab fitting function, we can get the variogram being $r(h) = 0.25 \times h^{1.68}$. Then, we can calculate the variogram between all known points. The value of sample RSSI points is related to the distance.

5.2.4. Evaluation Metrics. First, the signal propagation is simulated over the interest area. Meanwhile, the interpolation error of the Kriging interpolation algorithm is compared with nearest neighbor (NN) interpolation algorithm and inverse distance weighted (IDW) interpolation algorithm.

5.3. Prediction Performance Analysis. In this section, the performances of proposed method are evaluated. First, we show the results of user latent interest distribution. Then, the impact of interest number and the proportion of training set on link prediction can be verified.

5.3.1. Signal Propagation. When $\rho = 0.1$, 10% of all subareas of the target area are covered by the participants to collect WLAN RSSI values, and then we use all the WLAN data restored by the Kriging spatial interpolation algorithm according to 10% of the sample data of all data in the target

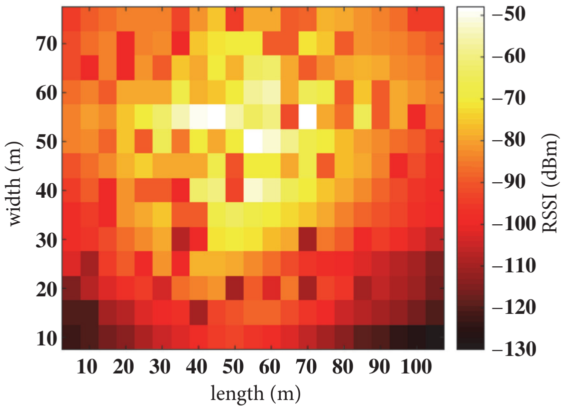


Figure 6: When ρ=0.1, using the Kriging interpolation algorithm restores the target area WLAN signal RSSI heat map.

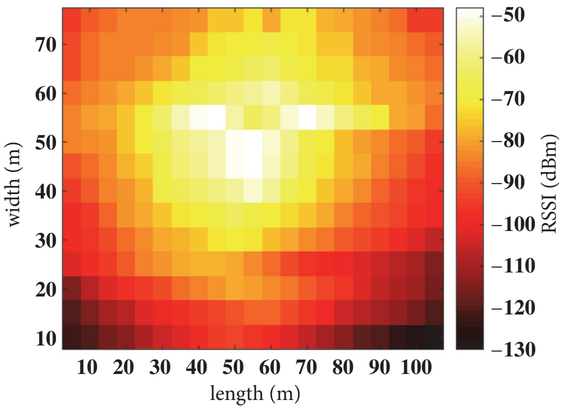


Figure 7: When ρ=0.3, using the Kriging interpolation algorithm restores the target area WLAN signal RSSI heat map.

area, as shown in Figure 6. The lighter color of the heat map indicates that the signal power of the WLAN received by smart phones at the location is greater, whereas the darker the heat map color, the smaller the WLAN power received by phones at the location. In Figure 6, it can be seen that the color in some areas is not smooth enough, and the RSSI data value fluctuates greatly and its continuity is poor. Figure 7 shows that 30% of all subareas of the target area are covered by the perceived user and collect valid sensory data. Then 30% of all the data in the target area are restored using the Kriging interpolation algorithm to restore the data of all WLANs in the target area. The color in the heat map is excessive. Comparing the two figures, we found that the WLAN RSSI value of the target area restored by spatial interpolation using 30% of all data using the target area is more accurate than the WLAN received power of the restored 10% of the target area using all the data.

5.3.2. Interpolation Algorithm Error Comparison. To verify the accuracy of the Kriging spatial interpolation algorithm, we used the nearest neighbor interpolation (NN) algorithm, inverse distance weighted (IDW) interpolation algorithm, and Kriging interpolation algorithm to predict the restoration goal under the same data volume of sample data.

Figure 8 shows the error comparison of the NN interpolation algorithm, IDW interpolation algorithm, and Kriging interpolation algorithm when the sample data occupies all the data in the target area. To verify the accuracy of different interpolation algorithms, we selected five subareas in the target area as comparison regions interpolated using different interpolation algorithms, and the positions of the five subareas are shown as the positions of the five blue stars in the target area, shown in Figure 8. Interpolation error calculation process is to choose to select each of the five subareas to use three different interpolation algorithms to calculate the estimated value of the point based on the nearby sample data, and then take the absolute value of the estimated value and the measured value of the point. Then the experiment is repeated several times to get the average value and the error using different interpolation algorithms, which can be expressed as follows:

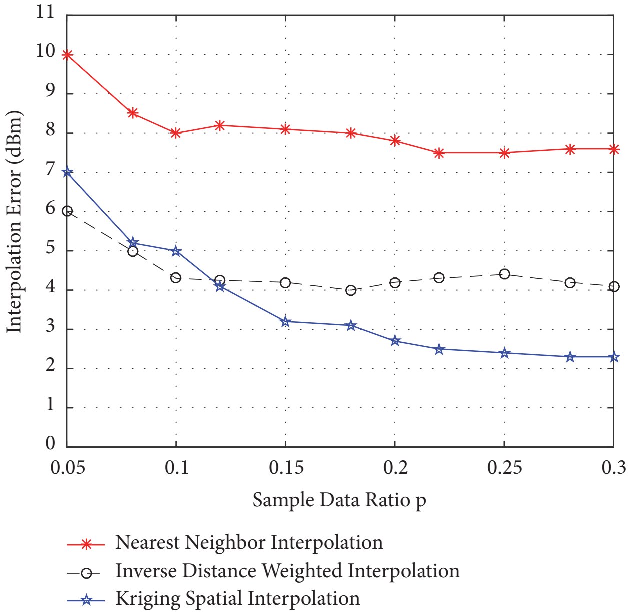


Figure 8: The interpolation error comparison of NN interpolation algorithm, IDW interpolation algorithm, and Kriging interpolation algorithm when the sample data proportion is different.

$$e = \frac{1}{5n}\sum_{i=1}^{n}\left|z_i - z^*\right|, \quad i = 1, 2, \ldots, n \tag{20}$$

where e denotes the error value after one of the interpolation algorithms uses multiple interpolations. n is the number of interpolation experiments using this interpolation algorithm. z_i indicates that this point uses some interpolation algorithm to obtain the estimated value based on the sample point data near the location. z^* is the measured data in the target area.

Figure 8 shows the error of different interpolation algorithms when the sample data occupies different proportions of the overall data. When the sample data occupies 0.05 of the total data, the interpolation error of the three difference

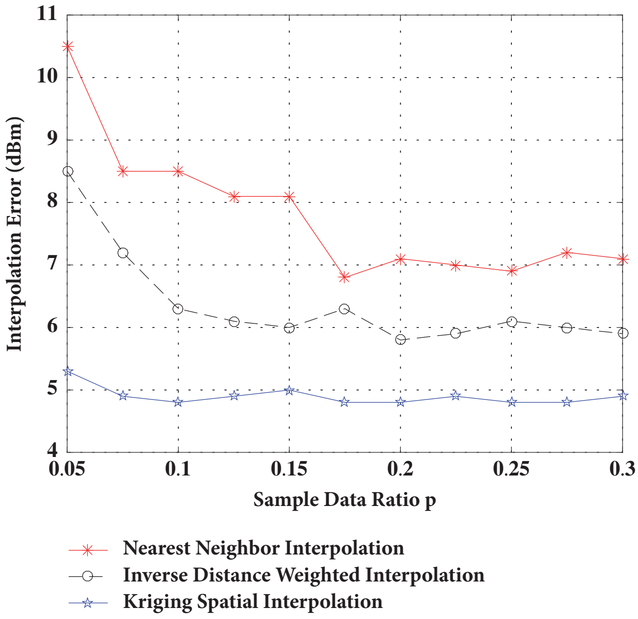


Figure 9: The interpolation error comparison of NN interpolation algorithm, IDW interpolation algorithm, and Kriging interpolation algorithm when the sample data proportion is different.

algorithms is relatively large. The interpolation errors of the nearest neighbor interpolation, Kriging spatial interpolation and inverse distance weighting are 10dBm, 7dBm, and 6dBm, respectively. With the increase of the proportion of sample data, the errors of the three interpolation algorithms are reduced. Among them, the error of Kriging spatial interpolation algorithm decreases sharply. Then the error of three interpolation algorithms tends to be stable, and it can be clearly seen that when the nearest neighbor interpolation algorithm is used, interpolation has the greatest error. Moreover, the IDW interpolation algorithm and Kriging spatial interpolation algorithm have smaller error when restoring the WLAN electromagnetic wave environment of the target area according to the sample data.

Figure 9 shows the values of the radio environment data obtained by interpolating sample data from the nearest five regions in a subregion of the target region using the nearest neighbor interpolation algorithm, inverse distance weighted interpolation algorithm, and Kriging interpolation algorithm. It can be seen from the figure that as the proportion of sample data increases; that is, the number of data samples increases; the errors obtained by using the three interpolation algorithms and the real data first decrease. Moreover, the NN interpolation algorithm has the largest error variation when the sample data is less than 0.2, the interpolation error of the IDW interpolation algorithm decreases as the sample data increases.

6. Conclusion

In this paper, we first introduced MCS to collect data for REM construction and proposed a system architecture to collect the radio environment information. Limited by the budget, only some of the participants can join the sensing task to collect the radio environment information, which leads to incomplete radio environment information. To solve the problem, the Kriging algorithm is proposed to infer the missing radio environment information data with collected sample radio environment information data. The performance is compared to the NN and IDW algorithms over different levels of sparsity. The simulation result shows that Kriging interpolation algorithm can infer the missing radio environment information data and generates more accurate radio environment information data than the NN and IDW algorithm. In the future, some data estimating and processing methods [24–26] can be used to add the sensing data for constructing the radio environment map.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was jointly supported by (1) National Natural Science Foundation of China (No. 61771068, 61671079) and (2) Beijing Municipal Natural Science Foundation (No. 4182041).

References

[1] P. Singh, M. Kumar, and A. Das, "Effective frequency planning to achieve improved KPI's, TCH and SDCCH drops for a real GSM cellular network," in *Proceedings of the 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT)*, pp. 673–679, India, July 2014.

[2] Z. Hou, Y. Zhou, L. Tian, J. Shi, Y. Li, and B. Vucetic, "Radio environment map-aided doppler shift estimation in LTE railway," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4462–4467, 2017.

[3] K. Ichikawa and T. Fujii, "Radio environment map construction using Hidden Markov Model in multiple primary user environment," in *Proceedings of the 2017 International Conference on Computing, Networking and Communications, ICNC 2017*, pp. 272–276, January 2017.

[4] K. Katagiri, K. Sato, and T. Fujii, "Crowdsourcing-Assisted Radio Environment Maps for V2V Communication Systems," in *Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, September 2017.

[5] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[6] B. Guo, Z. Wang, Z. Yu et al., "Mobile crowd sensing and computing: the review of an emerging human-powered sensing paradigm," *ACM Computing Surveys*, vol. 48, no. 1, article 7, 2015.

[7] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[8] J. Riihijarvi, J. Nasreddine, and P. Mahonen, "Demonstrating radio environment map construction from massive data sets," in *Proceedings of the 2012 IEEE International Symposium on Dynamic Spectrum Access Networks, DYSPAN 2012*, pp. 266-267, October 2012.

[9] S. Ulaganathan, D. Deschrijver, M. Pakparvar et al., "Building accurate radio environment maps from multi-fidelity spectrum sensing data," *Wireless Networks*, vol. 22, no. 8, pp. 2551–2562, 2016.

[10] S. Ureten, A. Yongacoglu, and E. Petriu, "A comparison of interference cartography generation techniques in cognitive radio networks," in *Proceedings of the ICC 2012 - 2012 IEEE International Conference on Communications*, pp. 1879–1883, June 2012.

[11] J. Liang, M. Liu, and X. Kui, "A survey of coverage problems in wireless sensor networks," *Sensors & Transducers*, vol. 163, no. 1, pp. 240–246, 2014.

[12] Z. Wei, Q. Zhang, Z. Feng, W. Li, and T. A. Gulliver, "On the construction of Radio Environment Maps for Cognitive Radio Networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 2013*, vol. 1, pp. 4504–4509, April 2013.

[13] V. Atanasovski, J. Van De Beek, A. Dejonghe et al., "Constructing radio environment maps with heterogeneous spectrum sensors," in *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN 2011*, vol. 3, pp. 660-661, May 2011.

[14] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 370, no. 1958, pp. 176–197, 2012.

[15] A. A. Khan, M. H. Rehmani, and A. Rachedi, "Cognitive-Radio-Based Internet of Things: Applications, Architectures, Spectrum Related Functionalities, and Future Research Directions," *IEEE Wireless Communications Magazine*, vol. 24, no. 3, pp. 17–25, 2017.

[16] A. Yaqot and P. A. Hoeher, "Efficient Resource Allocation in Cognitive Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6349–6361, 2017.

[17] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1340–1353, 2015.

[18] S. Grimoud, B. Sayrac, S. Ben Jemaa, and E. Moulines, "An algorithm for fast REM construction," in *Proceedings of the 2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications, CROWNCOM 2011*, pp. 251–255, June 2011.

[19] A. Umbert, F. Casadevall, and E. G. Rodriguez, "An outdoor TV band Radio Environment Map for a Manhattan like layout," in *Proceedings of the 13th International Symposium on Wireless Communication Systems, ISWCS 2016*, pp. 399–403, September 2016.

[20] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, 2014.

[21] R. Olivier and C. Hanqiang, "Nearest Neighbor Value Interpolation," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 4, 2012.

[22] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[23] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the ACM 23rd National Conference (ACM '68)*, pp. 517–524, 1968.

[24] Z. Ma, J. Xie, H. Li et al., "The role of data analysis in the development of intelligent energy networks," *IEEE Network*, vol. 31, no. 5, pp. 88–95, 2017.

[25] S. P. Bharati, F. Cen, A. Sharda, and G. Wang, "RES-Q: Robust Outlier Detection Algorithm for Fundamental Matrix Estimation," *IEEE Access*, vol. 6, pp. 48664–48674, 2018.

[26] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, "Decorrelation of neutral vector variables: theory and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 129–143, 2018.

Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 5650245, 13 pages
https://doi.org/10.1155/2019/5650245

Research Article

Enhanced Android App-Repackaging Attack on In-Vehicle Network

Yousik Lee,[1] Samuel Woo,[2] Jungho Lee,[3] Yunkeun Song,[1] Heeseok Moon,[4] and Dong Hoon Lee [5]

[1] *ESCRYPT, 4F, ABN Tower, 331 Pangyo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do 13488, Republic of Korea*
[2] *Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Republic of Korea*
[3] *Korea Information Certificate Authority Inc., 5th Fl C-dong, PDC, 242, Pangyo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do 13487, Republic of Korea*
[4] *KATECH, 74 Yongjeong-ri, Pungse-myeon, Dongnam-gu, Cheonan, Chungcheongnam-do, Republic of Korea*
[5] *Korea University, Anam-ro, Seongbuk-gu, Seoul 02841, Republic of Korea*

Correspondence should be addressed to Dong Hoon Lee; donghlee@korea.ac.kr

Received 18 October 2018; Accepted 27 December 2018; Published 3 February 2019

Guest Editor: Karl Andersson

The convergence of automobiles and ICT (information and communication technology) has become a new paradigm for the development of next-generation vehicles. In particular, connected cars represent the most in-demand automobile-ICT convergence technology. With the development of 5G technology, communication between vehicle and external device using autonomous driving and Internet of things (IoT) technology has been remarkably developed. Control of vehicles using smart phones has become a routine feature, and over 200 Android apps are in use. However, Android apps are easy to tamper by repackaging and allowing hackers to attack vehicles with using this vulnerability, which can lead to life-critical accidents. In this study, we analyze the vulnerabilities of connected car environments when connecting with IoT technologies and demonstrate the possibility of cyberattack by performing attack experiments using real cars and repackaging for commercial apps. Furthermore, we propose a realistic security technology as a countermeasure to attain safety against cyberattacks. To evaluate the safety of the proposed method, a security module is developed and a performance evaluation is conducted on an actual vehicle.

1. Introduction

Various electronic control units (ECUs) are being mounted in the latest vehicles to enhance the safety and comfort of the driver and passengers, and this automobile-ICT (information and communication technology) convergence has become a new paradigm for the development of next-generation vehicles [1, 2]. As the development of automotive electronics is accelerating, the proportion of electric parts in automobile production has exceeded 40% and is expected to exceed 50% by 2020. Along with the increase in automotive electronic parts, communications between vehicles and outside data services are also increasing. It is expected that around 75% of all vehicles worldwide will be connected cars using wireless networks by 2020 [3]. Connected cars, one of the leading use case of IoT, are growing rapidly with the development of 5G technology. [4] 5G will be the backbone of IoT, overcoming all space-time constraints and providing a complete connection with all the user-centric "things". [5] The existing V2X technology is evolving into C-V2X using LTE or 5G, and vehicles are connected to external devices via IoT technologies. Vehicles, which have for decades been typical machines, are evolving into large IT systems. As a result, IT experts can now participate in the development of vehicles and the general public can understand vehicles more easily.

However, with the development of automotive electronics, vehicles have become a new target for hackers [6–8]. Lee et al. published the results of a cyberattack experiment that analyzed the vulnerabilities of an ELM327-based CAN-to-Bluetooth device and apps for vehicles [9]. Furthermore, as shown in the hacking of GM OnStar by Samy Kamkar and the study on the vulnerabilities of in-vehicle smartphones by

Mikhail Kuzin and Victor Chebyshev, vulnerable smartphone apps are emerging as a new attack surface for vehicles [10, 11].

In the present study, we automate app analysis tasks that we performed manually in [9] and analyze the vulnerabilities of 213 apps registered in the Google Play store. Based on the results of this analysis, we carry out attacks to control actual vehicles after repackaging the 10 most popular apps among the commercial apps for vehicles. Finally, we propose a security technology to protect against such attacks and develop an access control device based on a whitelist for communication service security for ELM327 devices. These are universal access control devices applicable to all vehicles if CAN IDs are possible to be configured by car manufacturers.

This paper organized as follows. Background describes the overview of basic concepts and components. Next, Proposed Attack Model introduces how the attack model we proposed is constructed in both laboratory and real vehicle environment. Countermeasure describes the security measure against the attack we conducted. Finally, Conclusion presents our conclusions.

2. Background

2.1. CAN (Controller Area Network). To support efficient communication among ECUs, BOSCH developed the Controller Area Network (CAN) in the early 1980s. CAN is a sender ID-based broadcast communication technology that supports the bus network topology. CAN drastically decreased the complexity and length of in-vehicle communication lines by resolving the drawbacks of point-to-point communication. In the data transmission process between ECUs using CAN, the sending ECU includes its unique ID in the CAN data frame before sending the data. The receiving ECUs can selectively receive the data after checking the ID of the sending ECU included in the broadcast data frame. CAN bus systems are classified into the following two types depending on the length of the ID in the data frame:

(i) Standard CAN 2.0A (11-bit ID)

(ii) Extended CAN 2.0B (29-bit ID)

The data frame is composed of SOF (start of frame), Arbitration ID (arbitration), Control, Data, CRC (cyclic redundancy check), ACK (acknowledge), and EOF (end of frame) fields. The standard CAN 2.0A format uses an 11-bit ID as the arbitration bit, and the extended CAN 2.0B format uses a 29-bit ID as the arbitration bit.

Because CAN was designed only for closed network environments, it does not provide basic information protection features (e.g., confidentiality, authentication, and access control). Recently, as connected car services were commercialized, the use of FOTA (Firmware Over-The-Air) spread, in-vehicle networks, and external networks has become interconnected. As a result, CAN has been exposed to the same cyberattacks experienced in general IT environments.

2.2. Connected Car. The connected car environment implies that vehicles are always connected to an external network. The connected car environment consists of the following components [12, 13]:

(i) The vehicle in which ECUs have been installed and an in-vehicle network has been configured

(ii) The portal for providing various services to the vehicle

(iii) The communication link for connecting the vehicle with the portal

In such an environment, vehicles have multiple ECUs installed in them, which are connected to an in-vehicle network such as a CAN BUS system.

The communication link is constructed using a wireless communication device such as a telematics ECU or an ELM327. The portal is divided into web-based services and smartphone application-based services.

Through experiments, this study proves that vehicles are exposed to serious threats if connected car environments are constructed without solving the vulnerabilities of in-vehicle networks; a security mechanism is then proposed to solve this problem. The attack model that we propose has been designed based on the connected car environment shown in Figure 1.

Table 1: PID codes and parameters.

PID Code	Description	Units	Formula
04	Calculated engine load value	%	A∗100/255
0A	Fuel pressure	kPa	3∗A
0C	Engine RPM	kPa	(256∗A+B)/4
0D	Vehicle speed	Km/h	A

2.3. OBD Protocol. OBD (on-board diagnostics) is used to diagnose the vehicle state. The first OBD was produced to control the exhaust gases of vehicles [14]. As the development of automotive electronics accelerates, more ECUs have been installed and the OBD has developed to diagnose them and check the fault list. Later, OBD was established as an international standard (ISO 15765-4), and it has become possible to confirm the vehicle state with an automotive diagnosis device through standardized terminals [15]. In 1996, the US government mandated the installation of OBD-II in all vehicles sold in the US to control the environmental problems of exhaust gases.

The checking of vehicle status using OBD-II operates via the request/response method and the OBD PID (OBD Parameter ID) which is used in checking process defined in the SAE J1989 standard [16]. The structure of the CAN ID and data frame for communication defined in this standard is shown in Figure 2. The PID codes and parameters are defined in Table 1 [17, 18].

The diagnostic information of vehicles obtained through OBD-II can be easily acquired not only through the dedicated diagnostic devices used in car repair shops, but also through smartphone applications and ELM327 modules purchased from aftermarket suppliers. Therefore, malicious users can control vehicles by manipulating the diagnostic information

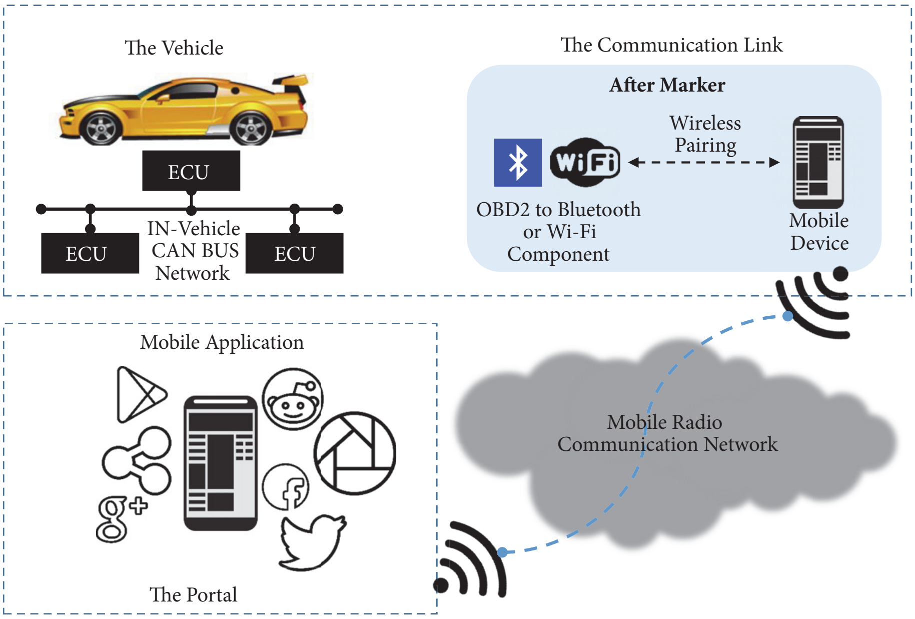
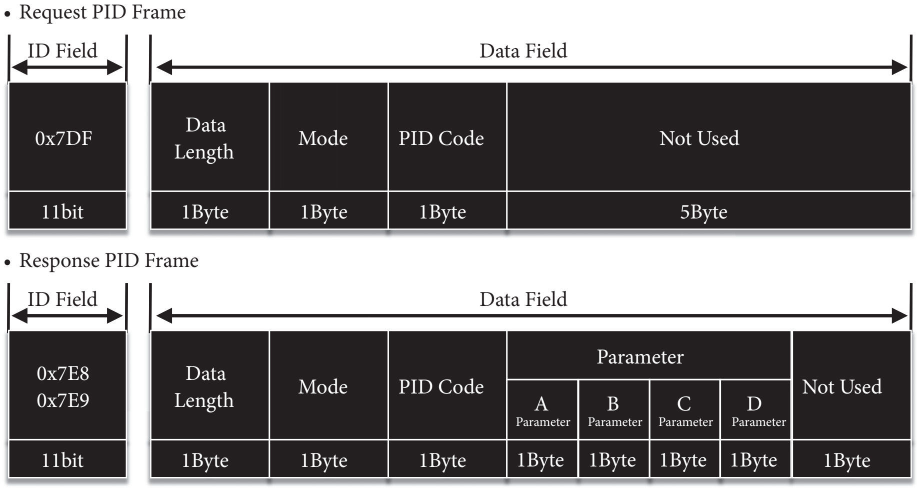


Figure 1: Connected car environment composed of ELM327 devices and mobile applications.



- Request PID Frame

ID Field	Data Field			
0x7DF	Data Length	Mode	PID Code	Not Used
11bit	1Byte	1Byte	1Byte	5Byte

- Response PID Frame

ID Field	Data Field							
0x7E8 0x7E9	Data Length	Mode	PID Code	Parameter				Not Used
				A Parameter	B Parameter	C Parameter	D Parameter	
11bit	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte	1Byte

Figure 2: Structure of CAN ID and data frame for communication.

of vehicles, and security measures are required for the diagnostic devices and related protocols.

2.4. ELM327. ELM327 is a microcontroller for car repair used to diagnose the vehicle state. If a wireless communication module such as Bluetooth is added to ELM327, the driver can check the vehicle state in real time by connecting his/her smartphone with ELM327. The process of checking the vehicle state using a smartphone and ELM327 is illustrated in Figure 3

2.5. Cyber Kill Chain. Lockheed Martin proposed a cyberattack process that is generally applicable to all cyberattacks [19]. They also proposed the Cyber Kill Chain method to identify the threats in each attack phase by analyzing the cyberattack process, and to increase the resilience of organizations by defeating and neutralizing the purpose, intention, and activities of attackers. The Cyber Kill Chain consists of seven steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. The characteristics of each phase are described in Table 2 [20].

3. Proposed Attack Model

The attack model we propose is more realistic than those of existing studies and has been designed on the basis of two threats. The first threat is that smartphone apps for

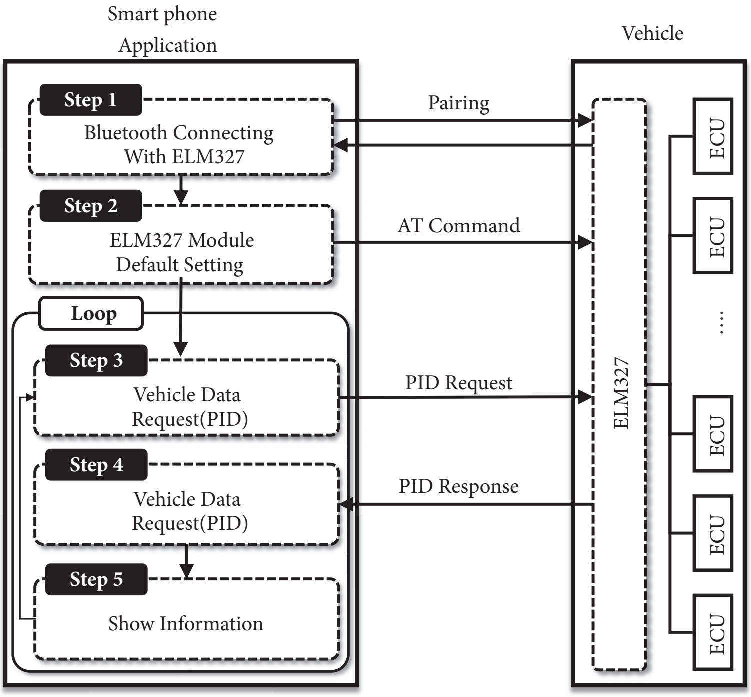


Figure 3: Process of checking vehicle status using smartphone and ELM327.

Table 2: Cyber Kill Chain phases and characteristics.

Phase	Action
Reconnaissance	Identify the target
Weaponization	Prepare the operation
Delivery	Launch the operation
Exploitation	Gain access to victim
Installation	Establish beachhead at the victim
Command and Control	Remotely control the implants
Actions on Objectives	Achieve the mission's goal

vehicles can be easily forged/repackaged and redistributed. The second threat is that attackers can attack wirelessly using repackaged malicious smartphone apps at any time. The proposed attack model is composed of an attacker, a target vehicle, and a victim. The characteristics of every entity are gathered to form one attack model. We analyzed the proposed attack model using the cyber intrusion kill chain.

3.1. Attacker Ability. Using a vehicle diagnostic device, the attacker can acquire a CAN data frame that can drive a specific ECU mounted in the target vehicle. Furthermore, the attacker can download a vehicle application that is distributed/sold in the app market and repackage it in a desired form. The attacker can then inject a desired CAN data frame into the in-vehicle CAN using a malicious app. The malicious app repackaged by the attacker can be distributed through a third-party market.

3.2. Target Vehicle. The target vehicle is assumed to utilize the connected car environment shown in Figure 1. The target vehicle uses CAN to facilitate communication among the ECUs. However, because the CAN bus system does not guarantee authentication for communicated data frames, a CAN data frame may be used in a retransmission attack [21, 22].

3.3. Victim Behavior. The driver of the target vehicle (victim) downloads a vehicle diagnosis app from the app market onto his/her smartphone and receives various services while driving his/her vehicle. The vehicle diagnosis app that the driver (victim) downloaded is assumed to be a malicious app distributed through the app market by the attacker. The driver (victim) using the malicious app cannot recognize the attack behaviors of the malicious app (injecting a data frame that can control a specific ECU).

3.4. Attack Model. The driver of the target vehicle (victim) who has downloaded the vehicle diagnosis app can receive various services while driving his/her vehicle. The attacker can inflict cyberattacks to many unspecified vehicles by abusing this service situation. The connected car environment in Figure 1 represents an environment in which a wireless communication module (ELM327) is installed in the OBD-II terminal. Through this terminal, the vehicle can connect to the in-vehicle CAN, and the smartphone app is always interconnected with the in-vehicle CAN during vehicle operation via the paring of the user smartphone and the ELM327 module. The attack model process is divided into

Table 3: Analysis of proposed attack model using Cyber Kill Chain.

Reconnaissance	(i) Analysis of vehicle control packet (using a diagnostic device) (ii) Analysis of an Android app for vehicles (iii) Analysis of ELM327 protocol
Weaponization	(i) Repacking the Android app for vehicles
Delivery	(i) Distribution to third-party app market
Exploitation Installation	(i) User downloads and installs the app on his/her smartphone
Command & Control	(i) Analysis of vehicle operation state through the in-vehicle network packets
Actions on Objectives	(i) Forced control of the automotive E/E system, causing a traffic accident

an attack preparation phase and an attack performance phase. The entire process including the attack preparation phase and attack performance phase is as follows:

(1) Attacker analyzes the communication vulnerabilities of a wireless communication module for vehicles (based on ELM327).

(2) Attacker downloads an app for vehicles from the app market, using a wireless communication module for vehicles based on ELM327.

(3) Attacker generates malicious code using the analysis results from step (1).

(4) Attacker inserts malicious code in the downloaded app for vehicles and repackages it.

(5) Attacker distributes the repacked malicious app through the app markets (including third-party markets).

(6) Victim installs the ELM327-based wireless communication module in the OBD-II terminal of his/her vehicle.

(7) Victim downloads the app for vehicles to his/her smartphone from the app market (or a third-party market) and installs it. It is assumed that the downloaded app is the malicious app distributed by the attacker.

(8) Victim runs the malicious app and drives his/her vehicle.

(9) As soon as the vehicle starts operating, the malicious app carries out a malicious act.

The attack model consists of nine steps in total as outlined above. Table 3 shows an analysis of the proposed attack model using the Cyber Kill Chain.

4. Attack Experiment

In this chapter, we prove the possibility of forced control of vehicles using an app repackaged by an attacker. Our attack experiment consists of two phases: preparation phases and actual attack phases. The preparation phase and the actual attack step are each divided into two detailed steps:

Table 4: AT commands for controlling vehicles.

AT Command	Description	Group
Z	Reset All	General
SH xyz	Set Header to xyz	OBD
AL	Allow Long(>7byte) message	ODB
AR	Automatic Receive	OBD
R0/R1∗	Responses Off/On∗	OBD
SP x	Set Protocol to h	OBD
CAF0/CAF1∗	CAN Automatic Formatting Off/On∗	CAN

(1) Preparation phase 1: communication protocol analysis of the CAN-to-Bluetooth module (ELM327 module).

(2) Preparation phase 2: vulnerability analysis of the vehicular application and production of the malicious vehicular application.

(3) Actual attack phase 1: LABCAR-based attack experiment.

(4) Actual attack phase 2: real car-based attack experiment.

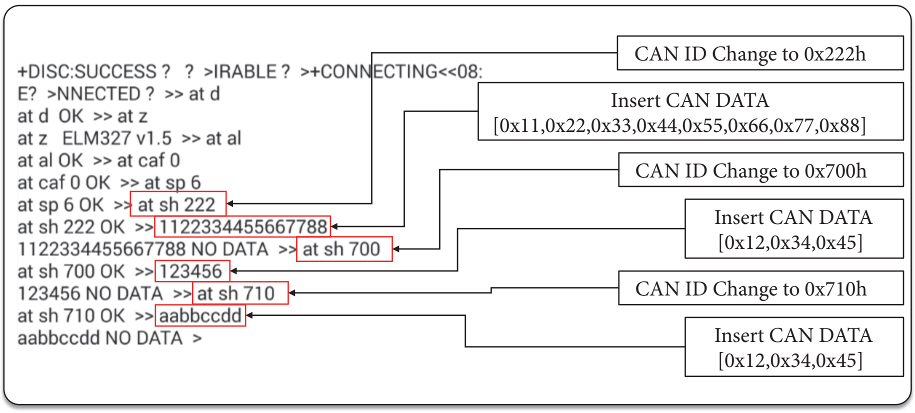
Finally, we performed a risk assessment for all apps related to ELM327 sold on the Google Store. In order to risk assessment, we manufactured automated analysis tools.

4.1. Preparation Phase (Analysis). In this section, the vulnerabilities of the ELM327 module and the vehicle diagnosis app are analyzed. In addition, all the apps sold/distributed in the Android app market are examined and the risk of app-repackaging attacks is analyzed.

(A) Analysis of Communication Protocol between Vehicle Diagnosis App and ELM327. As described in Section 2.4, the ELM327 module is mounted in the OBD-II port of the vehicle, and delivers the vehicle state information to the vehicle diagnosis app according to the request-response method. The ELM327 module generally uses a fixed CAN ID when sending a request message. However, in a special case, the CAN ID and data of the request message can be changed using the AT command provided by ELM Electronics. The AT commands that can be used to control vehicles are listed in Table 4.

However, even if the CAN ID and data are changed, the vehicle recognizes any data as normal data if it follows the communication protocols of AT commands and ELM327. In other words, if the CAN data desired by the user is sent to ELM327 as shown in Figure 4, the same information is received in the vehicle.

Every vehicle diagnosis app based on ELM327 acquires the vehicle state information using the OBD PID. Therefore, the OBD PID code can be easily found by decompiling and analyzing the vehicle diagnosis app. In the next section, we discuss the method of finding vulnerabilities using the AT command and OBD PID information in a commercial



```
+DISC:SUCCESS ?   ?  >IRABLE ?  >+CONNECTING<<08:
E?  >NNECTED ?  >> at d
at d  OK  >> at z
at z   ELM327 v1.5  >> at al
at al OK  >> at caf 0
at caf 0 OK  >> at sp 6
at sp 6 OK  >> at sh 222
at sh 222 OK  >> 1122334455667788
1122334455667788 NO DATA  >> at sh 700
at sh 700 OK  >> 123456
123456 NO DATA  >> at sh 710
at sh 710 OK  >> aabbccdd
aabbccdd NO DATA  >
```

CAN ID Change to 0x222h
Insert CAN DATA [0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88]
CAN ID Change to 0x700h
Insert CAN DATA [0x12,0x34,0x45]
CAN ID Change to 0x710h
Insert CAN DATA [0x12,0x34,0x45]

Figure 4: Message transmission from outside the vehicle to ELM327.

diagnosis app, and the method of repackaging it into a malicious app.

(B) Analyzing and Repackaging Vehicle Diagnosis App. For vulnerable vehicle diagnosis apps, the strings of the ELM327 AT command and OBD PID are exposed in plain text. If the attacker has decompiled the app, he/she can easily find the AT command and OBD PID by string search only. In this chapter, the method of searching attack points and inserting attack codes using the ELM327 AT command and OBD PID is described from the standpoint of an attacker, and the process of producing a malicious app using this method is explained.

Figure 3 shows the process in which the vehicle diagnosis app operates. The vehicle diagnosis app uses AT command and OBD PID after being connected with ELM327. This allows to analyze the vehicle diagnosis app easily.

When we decompile a target app for attack and search the AT command and OBD PID, they are searched. This searched string is then replaced with the vehicle control string desired by the attacker, and a malicious app is produced by repacking the app. Figure 5 shows how a desired vehicle control command can be inserted by simply modifying the searched AT command or the OBD PID.

The vehicle diagnosis app checks the vehicle status using the OBD PID. If the attacker can find the receiving location information in the decompiled vehicle diagnosis app, that location can be specified as the attack point. Figure 6 shows the results of finding the attack point and acquiring the vehicle information. If the attacker can insert the attack code for controlling the vehicle based on this information, he/she can arbitrarily control the vehicle. Figure 7 shows that the vehicle control command has been inserted at the desired point by the attacker.

Through the above analysis process, two attack tools can be produced. The attack tool types to be produced and the method of producing the attack tools are as follows:

(i) Delivery of attack command at the time the app is started when the vehicle diagnosis app is started; it initializes ELM327 using the AT command. Thus, the attacker replaces the AT command in the app to be attacked with the desired vehicle control command and repackages it.

(ii) Delivery of attack command in the specific condition of the vehicle.

The vehicle diagnosis app can check the vehicle status using the OBD PID. The attacker can insert a desired vehicle control command by using the repackaging method with the OBD ID in the above-mentioned analysis process.

In the next section, the forgery risk associated with the AT command and OBD PID of existing vehicle diagnosis apps distributed in the Android app market is analyzed.

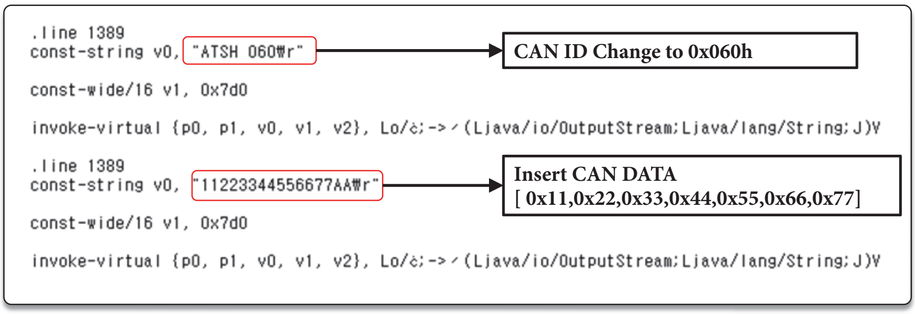
4.2. Actual Attack Experiment. This section describes the attack experiment, which utilizes an actual vehicle and the results from the preparation phase. The actual attack experiment in this section corresponds to steps (4) throw (7) of the seven steps of the Cyber Kill Chain. The attack experiment environment is outlined in Table 5.

(A) LABCAR. A LABCAR-based attack experiment was performed first to verify the effectiveness of the attack tools produced in the preparation phase. LABCAR is configured as shown in Figure 8, and each component has the functions described below.

If the attack is successful in the above-mentioned LABCAR environment, it is checked through the operation of the CANoe monitoring tool and the instrument panel. This LABCAR-based attack experiment confirmed that it is possible to inject a malicious data frame into an in-vehicle CAN through an app-repackaging attack.

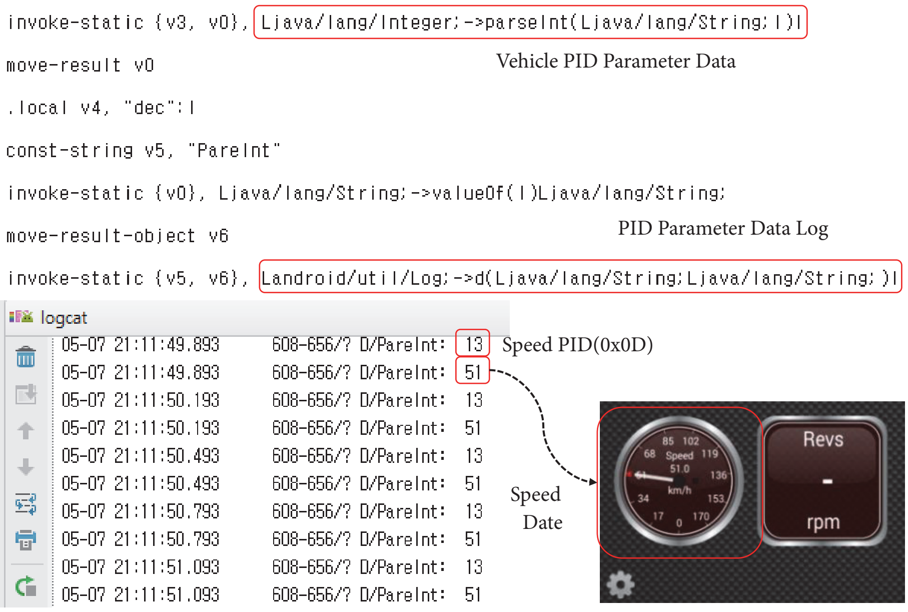
(B) Actual Vehicle. We also performed an attack experiment using an actual vehicle. The results of this experiment were recorded in a video, which was then uploaded to the web [23]. The attack experiment was carried out using the attack tools produced in the app analysis step. There were three attacks in total, which are described below.

(i) Door Unlock. After the driver leaves the vehicle in which ELM327 is installed, the attacker accesses the ELM327 through a smartphone and unlocks the door



```
.line 1389
const-string v0, "ATSH 060₩r"

const-wide/16 v1, 0x7d0

invoke-virtual {p0, p1, v0, v1, v2}, Lo/¿;->⁄(Ljava/io/OutputStream;Ljava/lang/String;J)V

.line 1389
const-string v0, "11223344556677AA₩r"

const-wide/16 v1, 0x7d0

invoke-virtual {p0, p1, v0, v1, v2}, Lo/¿;->⁄(Ljava/io/OutputStream;Ljava/lang/String;J)V
```

Figure 5: Message inserting CAN data by modifying AT command or OBD PID.



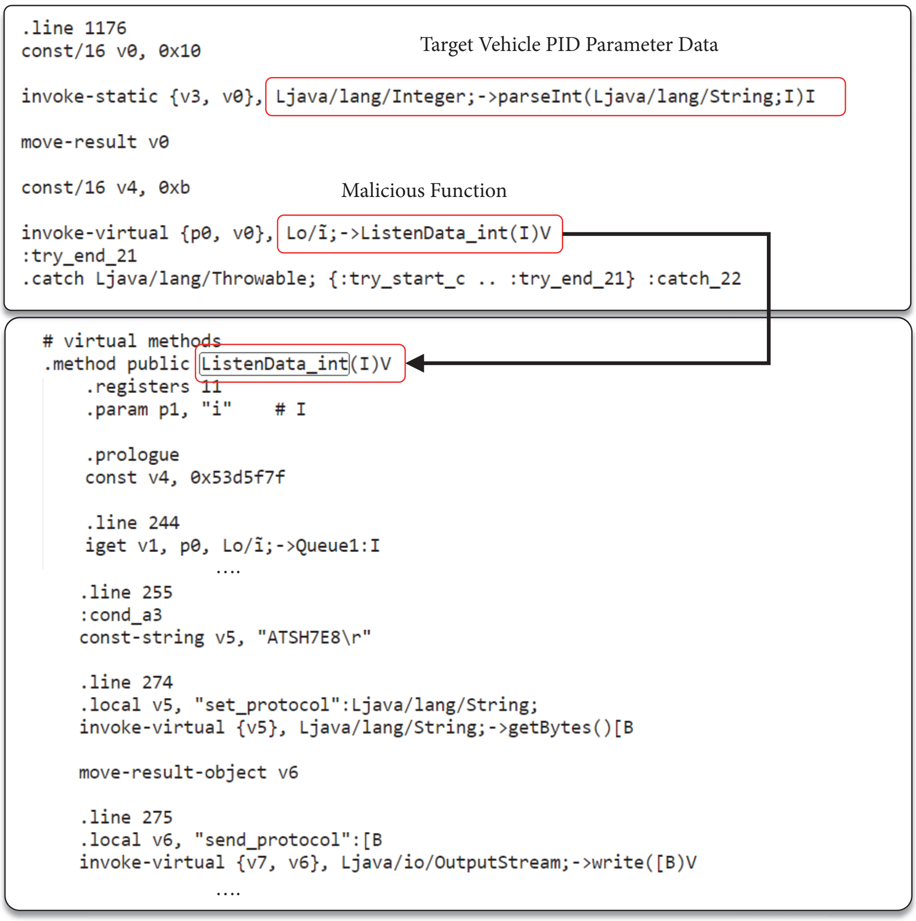
```
invoke-static {v3, v0}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;I)I

move-result v0

.local v4, "dec":I

const-string v5, "PareInt"

invoke-static {v0}, Ljava/lang/String;->valueOf(I)Ljava/lang/String;

move-result-object v6

invoke-static {v5, v6}, Landroid/util/Log;->d(Ljava/lang/String;Ljava/lang/String;)I
```

```
05-07 21:11:49.893     608-656/? D/PareInt:  13
05-07 21:11:49.893     608-656/? D/PareInt:  51
05-07 21:11:50.193     608-656/? D/PareInt:  13
05-07 21:11:50.193     608-656/? D/PareInt:  51
05-07 21:11:50.493     608-656/? D/PareInt:  13
05-07 21:11:50.493     608-656/? D/PareInt:  51
05-07 21:11:50.793     608-656/? D/PareInt:  13
05-07 21:11:50.793     608-656/? D/PareInt:  51
05-07 21:11:51.093     608-656/? D/PareInt:  13
05-07 21:11:51.093     608-656/? D/PareInt:  51
```

Figure 6: Message attack location of the analyzed vehicle diagnosis app and the acquisition of vehicle information.

Table 5: Cyber AT commands for controlling vehicles.

Experiment tools	Model Name	Functions & Features	Note
Vehicle	Omitted	(i) Target Vehicle (ii) Analyze CAN Data & Monitoring Target	Actual vehicle
Instrument panel	Omitted	(i) Testing Simulation ECU (ii) Target ECU	LABCAR actual vehicle
CAN BUS	CAN CASE XL	(i) Simulation CAN BUS (ii) Connecting ECU and CANoe	LABCAR
ELM327	ELM327 (Version 1.5)	(i) Diagnostic Device	LABCAR actual vehicle
Smartphone	Samsung Galaxy A3 (OS Version 5.1.1)	(i) Target Smartphone Device	LABCAR actual vehicle
Smali & BakSmali	Smali & BakSmali (Version 2.1.0)	(i) APK Assemble and Disassemble Tool	LABCAR actual vehicle
Signapk	Signapk	(i) APK Signing Tool	LABCAR actual vehicle
CANoe	CANoe (Version 7.1)	(i) CAN Monitoring Tool & Capture Tool	LABCAR actual vehicle
Diagnostic Application	Omitted	(i) Target Application (ii) Deployed Application	LABCAR actual vehicle



```
.line 1176
const/16 v0, 0x10

invoke-static {v3, v0}, Ljava/lang/Integer;->parseInt(Ljava/lang/String;I)I

move-result v0

const/16 v4, 0xb

invoke-virtual {p0, v0}, Lo/ĩ;->ListenData_int(I)V
:try_end_21
.catch Ljava/lang/Throwable; {:try_start_c .. :try_end_21} :catch_22
```

```
  # virtual methods
  .method public ListenData_int(I)V
      .registers 11
      .param p1, "i"    # I

      .prologue
      const v4, 0x53d5f7f

      .line 244
      iget v1, p0, Lo/ĩ;->Queue1:I
                   ....
     .line 255
     :cond_a3
     const-string v5, "ATSH7E8\r"

     .line 274
     .local v5, "set_protocol":Ljava/lang/String;
     invoke-virtual {v5}, Ljava/lang/String;->getBytes()[B

     move-result-object v6

     .line 275
     .local v6, "send_protocol":[B
     invoke-virtual {v7, v6}, Ljava/io/OutputStream;->write([B)V
                   ....
```

Figure 7: Vehicle diagnosis app in which the vehicle control command has been inserted at the desired point.

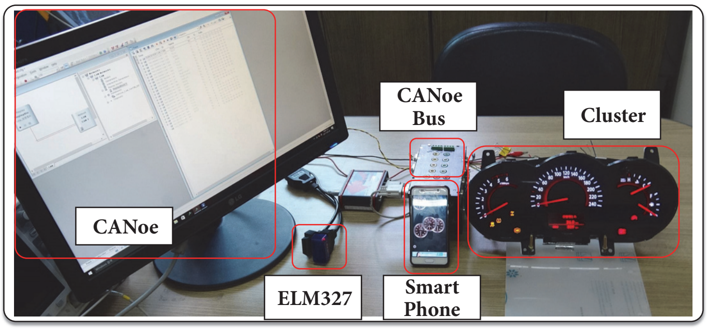


Figure 8: Operation LabCar simulation environment tool.

lock. Once the door lock is unlocked, the attacker can steal items from inside the vehicle and commit various other malicious acts.

(ii) Engine Off (Application Start). In this experiment, the Engine Off command set by the attacker is delivered when the repackaged app is started. The target vehicle was empty for safety and this experiment was performed with a safety device. It was confirmed that when the attacker installed a malicious app and started it, the Engine Off command was delivered and the target vehicle was stopped.

(iii) Engine Off (Vehicle's Status). In this experiment, the repackaged app delivered the Engine Off command in a specific state of the vehicle. It was confirmed that the engine stopped at the vehicle speed specified by the attacker.

Through the above three experiments, it was found that various apps sold/distributed in the Google Play store were

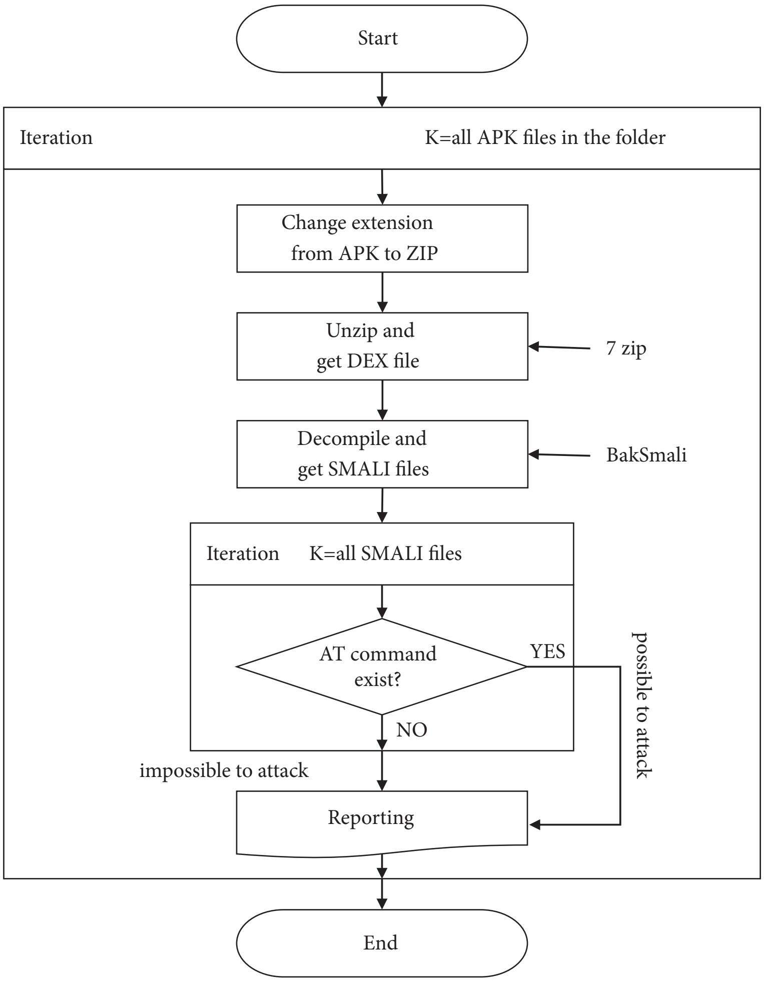
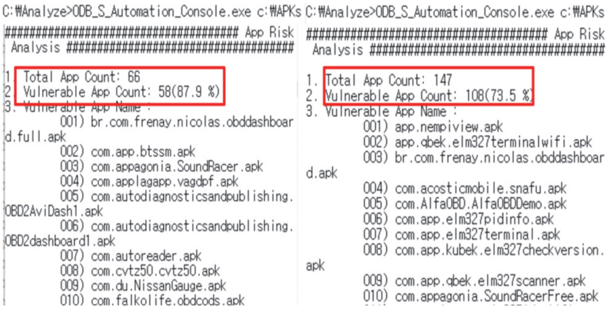


FIGURE 9: Operation process of the automatic vulnerability analysis.

exposed to the app-repackaging attack. There are primarily two reasons that the app-repackaging attack is possible. First, it is easy to analyze applications in terms of software. Second, the attack command can be delivered to the internal network through the OBD-II port of the vehicle in terms of hardware.

In the next chapter, we will discuss effective countermeasures against the above-mentioned attack.



```
C:₩Analyze>ODB_S_Automation_Console.exe c:₩APKs
########################################## App Risk
 Analysis ########################################

1. Total App Count: 66
2. Vulnerable App Count: 58(87.9 %)
3. Vulnerable App Name :
        001) br.com.frenay.nicolas.obddashboar
d.full.apk
        002) com.app.btssm.apk
        003) com.appagonia.SoundRacer.apk
        004) com.applagapp.vagdpf.apk
        005) com.autodiagnosticsandpublishing.
OBD2AviDash1.apk
        006) com.autodiagnosticsandpublishing.
OBD2dashboard1.apk
        007) com.autoreader.apk
        008) com.cvtz50.cvtz50.apk
        009) com.du.NissanGauge.apk
        010) com.falkolife.obdcods.apk
```

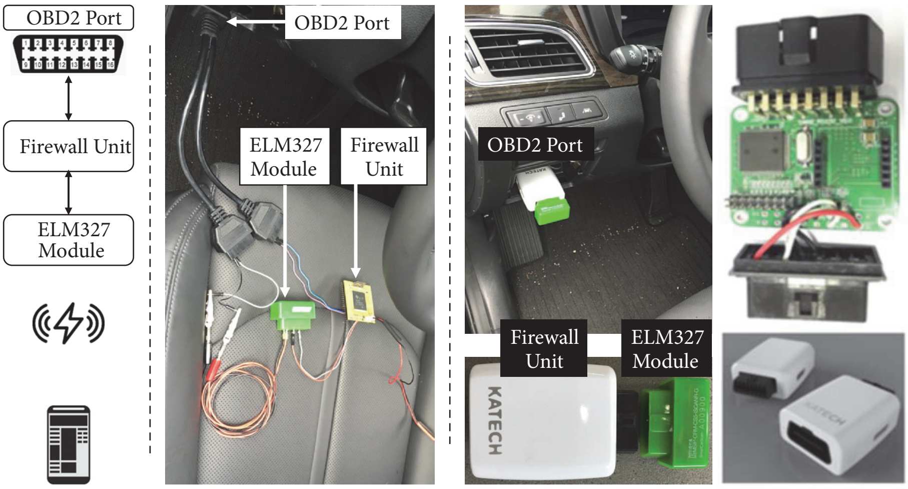
```
C:₩Analyze>ODB_S_Automation_Console.exe c:₩APKs
########################################## App Risk
 Analysis ########################################

1. Total App Count: 147
2. Vulnerable App Count: 108(73.5 %)
3. Vulnerable App Name :
        001) app.nempiview.apk
        002) app.qbek.elm327terminalwifi.apk
        003) br.com.frenay.nicolas.obddashboar
d.apk
        004) com.acosticmobile.snafu.apk
        005) com.AlfaOBD.AlfaOBDDemo.apk
        006) com.app.elm327pidinfo.apk
        007) com.app.elm327terminal.apk
        008) com.app.kubek.elm327checkversion.
apk
        009) com.app.qbek.elm327scanner.apk
        010) com.appagonia.SoundRacerFree.apk
```

FIGURE 10: Running screen of the automatic vulnerability analysis.

4.3. Analysis of the Risk of Commercial Vehicle Diagnosis Apps. In this section, the forgery risk associated with the AT command and OBD PID of apps is analyzed. The risk is analyzed by checking the existence of the AT command in the target app. In this study, the risk analysis was conducted for 213 apps related to ELM327 among the vehicle-related apps in the Google Play store (as of June 2017).

To check the existence of the AT command, the DEX (Dalvik Executable) file must be found first, which is a byte code-level executable file that exists in the APK (Android Application Package) file. Then the smali file is extracted by decompiling the DEX file, and the existence of the AT command is determined. The searching and analyzing process for checking for AT commands in several hundred apps can be automated because it is a mechanical, repetitive task [24]. In this study, an automatic vulnerability analysis tool was developed for this task [25]. Figure 9 shows the process used by the automatic vulnerability analysis tool, and Figure 10 shows the running screen.

The results of analyzing the apps using the automatic vulnerability analysis tool are outlined in Table 6. The number of apps that have a plain text AT command that can be



Firewall Unit Concept

(a)

Firewall Unit Prototype

(b)

Manufactured Firewall Unit

(c)

Figure 11: Access control system production process, (a) concept design, (b) prototype, and (c) actual manufacturing tool.

Table 6: App analysis results using automatic vulnerability analysis tool.

	Number of apps	Possible to attack (%)	Impossible to attack (%)
Total	213	166 (78)	47 (22)
Paid apps	66	58 (88)	8 (12)
Free apps	147	108 (73)	39 (27)

used for attack is 166, which accounts for approximately 78% of the 213 apps in total. In other words, most of the apps in the Google Play store can be used for attacks through app-repackaging. Furthermore, both free and paid apps are exposed to attacks.

The results of analyzing the apps using the automatic vulnerability analysis tool are outlined in Table 5. The number of apps that have a plain text AT command that can be used for attack is 166, which accounts for approximately 78% of the 213 apps in total. In other words, most of the apps in the Google Play store can be used for attacks through app-repackaging. Furthermore, both free and paid apps are exposed to attacks.

5. Countermeasure

In this chapter, we propose a security mechanism to protect against app-repackaging attacks. We designed a security mechanism considering the constraints of in-vehicle CANs and verified it. The proposed security mechanism is largely divided into two methods:

(i) Malicious data frame blocking method using a whitelist-based access control system

(ii) Code analysis defense method using app obfuscation

(A) Whitelist-Based Firewall. Most external devices such as ELM327 access the in-vehicle CAN through the OBD-II port. Car manufacturers must establish an access control policy for CAN data frames that are sent from external devices connected to the OBD-II port. In order to construct a safe in-vehicle CAN communication environment, a firewall is required to verify the ID of a CAN data frame (which comes in through the OBD-II port) and selectively send it to the subnetwork.

Car manufacturers can perform access control for CAN data frames that flow in through the OBD-II port during vehicle operation by generating a whitelist. The whitelist-based firewall can be designed/developed as follows:

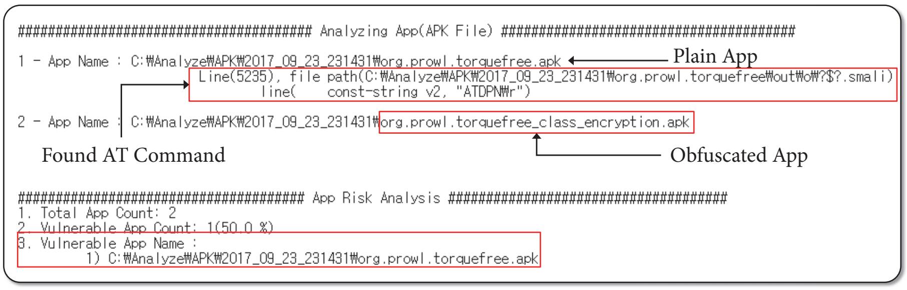
(1) Car manufacturer defines a CAN ID that can be flowed into the in-vehicle CAN through the OBD-II port during vehicle operation.

(2) Car manufacturer develops a whitelist-based firewall using the CAN ID defined in step (1) and installs it in the OBD-II port.

(3) Every external device can send data frames to the in-vehicle CAN only through the firewall installed at the OBD-II port.

(4) The firewall checks the data frames sent from external devices and drops abnormal data frames.

We designed and developed a whitelist-based firewall as shown in Figure 11. This firewall is installed between the OBD-II port and external device (ELM327 module) as shown in Figure 11(a). After producing the prototype shown in Figure 11(b), we connected it to an actual vehicle and performed an access control experiment. The prototype module was developed using an F28335-based microcontroller unit (MCU). All CAN data frames sent by ELM327 to the F28335-based MCU were checked first before being delivered to the



```
####################################### Analyzing App(APK File) #######################################

1 - App Name : C:₩Analyze₩APK₩2017_09_23_231431₩org.prowl.torquefree.apk
                       Line(5235), file path(C:₩Analyze₩APK₩2017_09_23_231431₩org.prowl.torquefree₩out₩o₩?$?.smali)
                               line(    const-string v2, "ATDPN₩r")

2 - App Name : C:₩Analyze₩APK₩2017_09_23_231431₩org.prowl.torquefree_class_encryption.apk

###################################### App Risk Analysis ######################################
1. Total App Count: 2
2. Vulnerable App Count: 1(50.0 %)
3. Vulnerable App Name :
        1) C:₩Analyze₩APK₩2017_09_23_231431₩org.prowl.torquefree.apk
```

Figure 12: Result of running the automatic vulnerability analysis tool for obfuscated app.

OBD-II port. Therefore, all data frames that do not have the predefined CAN ID could be dropped.

We developed the firewall device as a finished product with assistance from the Korea Automotive Technology Institute. The firewall device that we developed is shown in Figure 11(c). This firewall is equipped with an MCU based on an Infineon XC2265N (16-bit) microcontroller, and the whitelist can be easily updated. This access control policy for CAN data frames must be determined by the car manufacturer.

(B) App Obfuscation. The app-repackaging attack was possible because readability was provided for easy analysis of the internal app code, and the addition and repackaging of the app contents were possible. The app-repackaging attack is carried out largely in two steps.

In the first step, the AT command of the target app is found. In the second step, the app is repackaged by inserting malicious code. From the viewpoint of the Cyber Kill Chain, if only one of the attack steps is disabled, the attack cannot be performed.

Practically, the most effective method for preventing an app-repackaging attack is to prevent the AT command from being found.

The obfuscation technique that is mainly applied in Java is described in Table 7 [26].

When class encryption is not utilized, it is difficult to defend against attacks that find the AT command through string encryption, renaming, control flow, and API hiding.

Figure 12 shows the analysis result obtained using the automatic vulnerability analysis tool developed in this study, in a case where analyzing source code inside the app is made difficult by applying the class encryption method, which encrypts the DEX file and dynamically loads it when running the app.

In order to defend against the attacks performed in this study, it is recommended to apply class encryption, which makes the AT command unsearchable by encrypting the DEX file that includes the original source code.

6. Related Studies

K. Kosher et al. first reported ECU forced control attacks using actual vehicles. CAN data frames that can control vehicles by force using CAN message fuzzing, ECU firmware reverse engineering, and vehicle diagnostic device analysis were obtained. Using the acquired CAN data frames, wired and wireless attack models were proposed and actual vehicle hacking experiments were conducted. Among the proposed attack models, the short-range wireless attack and the long-range wireless attack showed very threatening results. It has been proven that a short-range wireless attack can hack vehicles in a Bluetooth pairing between a smartphone infected with malicious code and the hands-free function of vehicles. Furthermore, it has been experimentally proven that long-range wireless attacks can be performed using the vulnerabilities of the communication protocol and the telematics devices installed in vehicles [7].

M. Wolf, T. Hoppe, and others proved through an experiment based on the CANoe simulator that a message retransmission attack is possible in an in-vehicle CAN. To solve this problem, they proposed a message authentication method based on certificate-based ECU authentication and symmetric keys. However, it is impossible to apply their proposed security mechanisms to an actual vehicle environment because the computing power of ECUs and the payloads of CAN data were not considered [27, 28].

D. K. Nilsson et al. proposed the DDA (Delayed Data Authentication) method, which considered the computing capacity of ECUs and the limited CAN message structure. The DDA method uses the message authentication code (MAC) to prevent message retransmission attacks. They proposed a method of using the CRC field, while pointing out the shortage of areas available in the CAN message structure. Furthermore, they proposed a DDA-based method of authentication in which four messages are grouped. However, the CRC field in CAN cannot be used dynamically. In addition, real-time data processing must be guaranteed in vehicles. It is impossible to apply the DDA method to the vehicle environment because it generates a delay of at least 80 ms in authentication [29].

Woo et al. proposed a data frame authentication method using 32-bit truncated MAC. In consideration of the limited characteristics of the CAN data frame, they proposed a method of using the CRC and Extended ID fields for message authentication. In addition, they proposed a method of updating the session key to strengthen the safety of the 32-bit truncated MAC. This method ensures real-time data processing without generating additional data frames. However, it is impossible to apply their method to an actual

Table 7: App analysis results using automatic vulnerability analysis tool.

Obfuscation	Description
string encryption	The used string is replaced with an encrypted string, and a decryption method is added to the class file and the encrypted string is decrypted during runtime.
renaming	The classes, fields, and methods are renamed with meaningless names to make it difficult to analyze the decompiled source code.
control flow	The positions of commands in the code area of the class file are changed or trash commands are inserted to make it difficult to analyze flow during decompiling.
API hiding	Sensitive libraries are used or the method calling is hidden.
class encryption	A specific class file is encrypted and stored, and the dynamically decrypted code is run during runtime.

vehicle environment because the CAN protocol itself must be modified in order to use the CRC field for sending the message authentication code [6].

7. Conclusion

In this study, we have shown that vehicles can be easily attacked by analyzing ELM327s, which are commercial products that send vehicle information to smartphones and examining Android apps in the app market (Google Play store) that show vehicle information to owners after receiving the information. As countermeasures to such attacks, app obfuscation and a whitelist-based firewall were proposed. For the attack method used in this study, commercial products that are now sold or distributed were used. Because anyone can repackage Android apps if they have abilities to develop and analyze them, this has large ripple effects in terms of reality and risk. The more the 5G is expanded, the more the connection between the vehicle and the external device is increased; thus, the risk of cybersecurity will be increased. Therefore, multistep, multifaceted security measures are required to attain security against realistic threats. In the case of the firewall proposed in this study, a separate device that operates based on a whitelist is attached to an aftermarket device. However, car makers must install the security feature in the OBD-II interface, because we cannot realistically depend on an external device to counter threats to the vehicle itself. Furthermore, from the Cyber Kill Chain perspective, the security measures proposed in this study are countermeasures in terms of "reconnaissance" and "command and control." The security measures in the command and control step are still effective because using Wi-Fi instead of Bluetooth for the communication between the OBD-II device and smartphone apps, using another dongle instead of ELM327, or using iOS instead of Android is in the "reconnaissance," "identification," or "vulnerability" steps. In the "Actions on Objectives" step, the attacks can be detected and defended using IDS. Research on this topic is left for future studies.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Korea Ministry of Land, Infrastructure and Transport. It was also supported by the Korea Agency for Infrastructure Technology Advancement (Project no. 17TLRP-B117133-02).

References

[1] A. Saad and U. Weinmann, "Automotive Software Engineering and Concepts," *GI Jahrestagung*, vol. 34, pp. 318-319, 2003.

[2] E. Nickel, "IBM Automotive Software Foundry," in *Proc. Conf. Comput. Sci. in Automot. Ind.*, Frankfurt University, Frankfurt, Germany, 2003.

[3] J. Greenough, "The Connected Car Report: Forecasts, Competing Technologies, and Leading Manufacturers," Business Insider, 2016, http://www.businessinsider.com/connected-car-forecasts-top-manufacturers-leading-car-makers-2015-3.

[4] S. Pandi, F. H. P. Fitzek, C. Lehmann et al., "Joint design of communication and control for connected cars in 5G communication systems," in *Proceedings of the 2016 IEEE Globecom Workshops, GC Wkshps 2016*, December 2016.

[5] J. Ni, X. Lin, and X. S. Shen, "Efficient and Secure Service-oriented Authentication Supporting Network Slicing for 5G-enabled IoT," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 644–657, 2018.

[6] S. Woo, H. J. Jo, and D. H. Lee, "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015.

[7] K. Koscher, A. Czeskis, F. Roesner et al., "Experimental Security Analysis of a Modern Automobile," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, pp. 447–462, Oakland, CA, USA, May 2010.

[8] C. Miller and C. Valasek, "Demo: Adventures in Automotive Networks and Control Units," in *Proceedings of the DEFCON 2013*, 2013.

[9] J. H. Lee, S. Woo, S. Y. Lee, and D. H. Lee, "A Practical Attack on In-Vehicle Network Using Repackaging Android Applications," *Journal of the Korea Institute of Information Security and Cryptology*, vol. 26, no. 3, pp. 679–691, 2016.

[10] S. Karmar, "Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars," in *Proceedings of the DEFCON 23*, 2015.

[11] M. Kuzin and V. Chebyshev, "Hey Android, Where is My Car?" in *Proceedings of the RSA Conference 2017*, 2017.

[12] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 528–533, Baden-Baden, Germany, June 2011.

[13] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Creating a Secure Infrastructure for Wireless Diagnostics and Software Updates in Vehicles," in *Proc. Conf. Comput. Saf., Reliab., and Secur.*, pp. 207–220, Newcastle upon Tyne, UK, 2008.

[14] A. Tahat, A. Said, F. Jaouni, and W. Qadamani, "Android-based universal vehicle diagnostic and tracking system," in *Proceedings of the 2012 IEEE 16th International Symposium on Consumer Electronics (ISCE 2012)*, pp. 137–143, Harrisburg, Pa, USA, June 2012.

[15] The International Organization for Standardization (ISO), "The International Organization for Standardization (ISO)," The International Organization for Standardization standard 15765-4:2016, 2016.

[16] Society of Automotive Engineers (SAE) International, "Recommended Service Procedure for the Containment of Cfc-12," Society of Automotive Engineers standard J1989, 2011.

[17] S. Chen, J. Pan, and K. Lu, "Driving Behavior Analysis Based on Vehicle OBD Information and AdaBoost Algorithms," in *Proceedings of the in International MultiConference of Engineers and Computer Scientists (IMECS) 2015 Vol I*, pp. 102–106, 2015.

[18] C. Furmanczyk, D. Nufer, B. Sandona et al., *Integrating ODB-II, Android, and Google App Engine to Decrease Emissions and Improve Driving Habits*, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.469.359&rep=rep1&type=pdf.

[19] Lockheed Martin Corporation, "Seven Ways to Apply the Cyber Kill Chain® with a Threat Intelligence Platform," white paper, 2015.

[20] Lockheed Martin Corporation, "Gaining the Advantage: Applying Cyber Kill Chain® Methodology to Network Defense," Guide, 2015.

[21] T. Hoppe and J. Dittman, "Sniffing/Replay Attacks on CAN Buses: A Simulated Attack on the Electric Window Lift Classified Using an Adapted CERT Taxonomy," in *Proc. Conf. on Embed. Syst*, 2007.

[22] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures," *Reliability Engineering & System Safety*, vol. 96, no. 1, pp. 11–25, 2011.

[23] 2017, https://sites.google.com/team-aegis.com/webpage/publish_data.

[24] F. Palmieri, U. Fiore, and A. Castiglione, "Automatic security assessment for next generation wireless mobile networks," *Mobile Information Systems – Emerging Wireless and Mobile Technologies*, vol. 7, no. 3, Article ID 404328, pp. 217–239, 2011.

[25] GitHub repository, 2017, https://github.com/song4jang/ODB-S-app_repackaging-.

[26] Y. Piao, J. Jung, and J. H. Yi, "Structural and functional analyses of proguard obfuscation tool," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 38, no. 8, pp. 654–662, 2013.

[27] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the art: Embedding security in vehicles," *EURASIP Journal on Embedded Systems*, vol. 2017, 2007.

[28] T. Hoppe and J. Dittman, "Sniffing/replay attacks on can buses: A simulated attack on the electric window lift classified using an adapted cert taxonomy," in *Conf. Embedded Syst. Security*, pp. 1–6, 2007.

[29] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes," in *Proceedings of the 2008 IEEE 68th Vehicular Technology Conference (VTC 2008-Fall)*, pp. 1–5, Calgary, Canada, September 2008.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7943586, 13 pages
https://doi.org/10.1155/2018/7943586

Research Article

Automatically Traceback RDP-Based Targeted Ransomware Attacks

ZiHan Wang, [1] ChaoGe Liu, [1] Jing Qiu, [2] ZhiHong Tian, [2] Xiang Cui, [2] and Shen Su [2]

[1] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[2] *Cyberspace Institute of Advanced Technology Guangzhou University, Guangzhou, China*

Correspondence should be addressed to ChaoGe Liu; liuchaoge@iie.ac.cn, Jing Qiu; qiujing@gzhu.edu.cn, and ZhiHong Tian; tianzhihong@gzhu.edu.cn

Received 13 July 2018; Revised 24 October 2018; Accepted 22 November 2018; Published 6 December 2018

Guest Editor: Vishal Sharma

While various ransomware defense systems have been proposed to deal with traditional randomly-spread ransomware attacks (based on their unique high-noisy behaviors at hosts and on networks), none of them considered ransomware attacks precisely aiming at specific hosts, e.g., using the common Remote Desktop Protocol (RDP). To address this problem, we propose a systematic method to fight such specifically targeted ransomware by trapping attackers via a network deception environment and then using traceback techniques to identify attack sources. In particular, we developed various monitors in the proposed deception environment to gather traceable clues about attackers, and we further design an analysis system that automatically extracts and analyze the collected clues. Our evaluations show that the proposed method can trap the adversary in the deception environment and significantly improve the efficiency of clue analysis. Furthermore, it also helps us trace back RDP-based ransomware attackers and ransomware makers in the practical applications.

1. Introduction

Ransomware was first emerged in late 1980s [1, 2] and has resurfaced since 2013 [3]. Recently, several wide-spread ransomware attacks have caused significant damages on a large number of user systems and businesses on the Internet. *Symantec* reported a 250% increase in new crypto ransomware families between 2013 and 2014 [2]. In May 2017, *WannaCry* spread across more than 150 countries and 200,000 computers in just a few days, and severely disrupted many businesses and personal systems [4, 5]. In addition, *specifically targeted ransomware* like *Crysis* disrupted many small and large enterprises across the globe; e.g., *Trend Micro* observed that the *Crysis* family specifically targeted businesses in Australia and New Zealand in September 2016. The number of such targeted ransomware attacks was doubled in January 2017, compared with in late 2016 [6]. What is more, the lack of focus on security has left IoT (Internet of Things) devices vulnerable, which has been the target of 10% of all ransomware attacks. Researcher predicts IoT ransomware attacks being likely to increase to around 25% to 30% of all ransomware cases [7].

Because traditional ransomware was typically spread randomly without specific targets via network scanning or host probing, they can be easily detected by monitoring of the abnormal behaviors in host activities such as file system operations and network traffic [1, 3, 8]. Recently, more and more ransomware attacks aimed at specific targets. *Kaspersky Security Bulletin* indicated that targeted attacks have become one of the main propagation methods for several widespread ransomware families in 2017 [9, 10]. For instance, an attacker using *Crysis* ransomware first logs in a victim's host and spreads itself via a brute force attack on the common Remote Desktop Protocol (RDP). Such a targeted ransomware attack usually has a clear command-and-control structure and aimed at resource exploitation and resource theft on these targets, while generating fairly limited noisy on hosts and networks which is hard to detect.

Existing ransomware defense methods (designed for dealing with randomly-spread attacks) usually protect a host by blocking the spreading of ransomware attacks (in nearly real-time) based on the signatures generated by ransomware detection solutions. However, because of the different characteristic of targeted ransomware attacks with less notable patterns, these traditional blocking-based defense systems become much less effective for these targeted attacks.

To address this issue, we propose to utilize advanced defense schemes to protect important hosts under targeted ransomware attacks. In this paper, we utilize the *cyber deception technology* to help us protect critical systems through attack guidance, by drawing attackers away from these protected systems. While the cyber deception technology helps us protect important targets (such as in dealing with the Advanced Persistent Threat (APT) [11, 12]), it cannot help us traceback attack sources. To address this issue, we further design specific techniques to traceback RDP-based ransomware attacks and identify the original attack sources as the main deterrence of ransomware attackers.

Our deception environment simulates an actual user system in three layers with multiple monitors to observe various key system operations, related to login, network communication, clipboard, process, shared folder, and file system. It collects traceable clues and helps us detect the RDP ransomware attack. Because traditional traceback methods usually require security experts to manually analyze a large amount of collected clues, it is difficult for make them to achieve fast responses. Therefore, we develop an *automatic analysis system* to work on traceable clues by taking advantage of natural language processing and machine learning techniques.

To evaluate our system, we invite 122 volunteers in a simulated RDP-based ransomware attack. The proposed system was able to capture traceable clues through the proposed deception environment. It can also automatically analyze the clues effectively. The convergence rate of the analysis system reaches about 98%. Moreover, we demonstrated that it helps us traceback RDP-based attack sources in practical applications.

In summary, this paper makes the following contributions:

(i) We propose a systematic method to deter RDP-based ransomware by identifying attackers, which traps ransomware attackers via a cyberdeception environment and uses an automatic analysis system to obtain traceable clues and identify attack sources.

(ii) We build a deception environment to trap RDP-based ransomware attacker, by simulating an user environment in three layers: a network layer, a host layer and a file system layer. The environment helps us discover attacker behaviors and collects attacker-related information.

(iii) We develop an automatic analysis system with natural language processing and machine learning techniques to automatically recognize effective clues for tracing back ransomware attack sources.

(iv) We designed two practical experiments to test RDP-based ransomware attacks and ransomware makers, and demonstrated the feasibility of the proposed system.

The remainder of the paper is structured as follows. In Section 2, we briefly present background and related work. In Section 3, we describe the methodology of our systematic method. In Section 4, we present the implementation of our deception environment prototype. In Section 5, we describe the details of the clue analysis system. In Section 6, we discuss the evaluation setup and results. We conclude this paper in Section 7.

2. Background and Related Work

2.1. Related Work on Ransomware Defense. Ransomware is a type of malware which manipulates an user system to extort money. It operates in many different ways, e.g., simply locking a user's desktop or encrypting an entire file system. Recent rampant ransomware attacks have called for effective ransomware defense solutions. In the studies that tackle ransomware counteraction, several solutions are proposed to confront this attack [14, 15].

Some of these solutions are proposed to deal with all types of ransomware [1, 16–20]. For example, *Kharraz* presented a dynamic analysis system called *UNVEIL*. The system analyzes and detects ransomware attacks by modeling ransomware behaviors. It focuses on the observation of three elements, namely, I/O data buffer entropy, access patterns and file system activities [1]. Moreover, some others are type-specific solutions that deal with only one type, such as crypto-ransomware [21–25]. For example, *Scaife* presented an early-warning detection system that alerts users during suspicious file activities [21]. Utilizing a set of behavior indicators, the detection system can halt a process that appears to tamper with a large amount of user data. Furthermore, it is claimed that the system can stop a ransomware execution with a median loss of only 10 files. Similarly, some studies tackle the detection of specific ransomware families only [26–28]. For example, *Maltester* is a family-specific technique proposed by *Cabaj* to detect *Cryptowall* infections [27]. It employs dynamic analysis along with honeypot technology to analyze the network behavior and detect the infection chain.

These solutions can be categorized into prevention and detection. However, these two kinds of countermeasures have the following disadvantages. First, many prevention measures require many services to be disabled, which is likely to affect service functionality. For example, *Prakash* suggested several prevention measures including disabled macros in office documents, and restricted access permissions on "Temp" and "Appdata" folders [29]. Secondly, the detecting system is often difficult to conceal itself and perform its functions when against ransomware attacks that precisely aiming at specific hosts, e.g., using the common Remote Desktop Protocol (RDP). Finally, while these countermeasures can be used to detect or block specific ransomware attacks, they cannot fundamentally inhibit the spread of ransomware. But traceability technology can fundamentally inhibit the ransomware spread by traceback to attack sources.

2.2. RDP-Based Ransomware Attacks. Traditional ransomware randomly spreads across the Internet, in executable files, development kits, macro files, and other malicious programs on a large scale with various dissemination methods, including phishing emails, puddle attacks, vulnerability attacks, server intrusion, and supply chain pollution. They

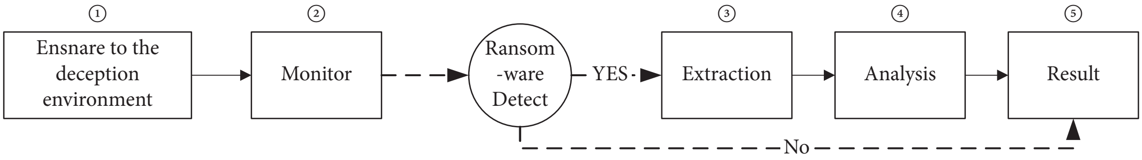


Figure 1: Data collection and analysis process of the whole prototype.

use different ways to trick a victim to launch such programs. Among these dissemination method, phishing emails is the most widely used. However, according to Kaspersky's 2017 ransomware report, the number of targeted ransomware attacks based on RDP is growing rapidly.

Recently, more and more ransomware criminals have spread ransomware using RDP services and then installed ransomware manually. These attackers use a brute-force method to acquire usernames and passwords on a target machine with an active RDP service [9]. For instance, one of the typical families, *Crysis*, a copycat of *Locky*, not only aims at common businesses but also targets healthcare service providers [9]. *Crysis* gains access to admin level privileges by stealing passwords and credentials. In addition, during an RDP session, the attacker uses both clipboard and shared folders to upload files to a remote host. And then, attackers can installed ransomware manually.

2.3. Cyber Deception Technology. Because many critical systems are known and always on, it is difficult to protect them from potential network attacks.

(i) Attackers can use zero-day vulnerabilities, highly antagonistic malicious code, or other resources to break the defense system.

(ii) Because humans are always the weakest link in defense systems, attackers can use social engineering to identify system weaknesses and penetrate the defense.

(iii) Attackers can repeatedly explore the potential vulnerabilities on a target system to identify its weaknesses.

However, when an attacker aims at a specific target, e.g., exploiting its RDP service, traditional passive defense methods cannot be usually less effective. Therefore, we need to use advanced active solutions to deal with such attacks with less observable features, such as cyber deception.

The earlier use of cyber deception technology is honeypot. Honeypot detects attacks by deploying a series of systems or resources in the service network that do not have real business. When a trap is accessed, it represents an attack. Honeypot system generally waiting attacks passively, and does not have the role of misleading and confusing attackers. What's more, the honeypot system does not have real business, and does not have high interactive characteristics, which may easy to be identified by attackers. Compared with the traditional honeypot system, the cyber deception system can be deployed more conveniently, the cyber deception environment is more real, and can be linked with existing defense products. It can provide more effective solutions for APT attacks, ransomware attacks, intranet attacks and other threats defense. A Gartner report in 2015 [30] pointed out the market prospect of deception-based security defense technology and predicted that 10% of organizations will use deception tools (or tactics) to counter cyber-attacks in 2018. Compared with the traditional passive defense approach, cyber deception technology is an active defense approach and can be applied to all stages of network attacks. We can use this technology to trap the RDP-based attacker, detect targeted attacks, and deter ransomware attackers by precisely identifying them.

Trap RDP-Based Ransomware Attackers. A targeted ransomware attack generally has three steps: detection, infiltration, and execution [31]. However, traditional security solutions are unable to cope with the internal translation phase. In addition, traditional honeypot technology (often used to fight network attacks) generally does not focus on tracing back to attackers. However, cyber deception technology can deceive the attacker into a surveillance environment and consume his time and energy with bait information.

Detect RDP-Based Ransomware Attacks. Once the attacker obtains the correct username and password combination, he usually returns multiple times within a short period to try and infect the compromised host [6]. In one particular case, *Crysis* was deployed six times on an endpoint within a span of 10 minutes. As a result, by monitoring in the cyber deception environment, we can detect RDP-based ransomware attacks in time and determine the attacker's behavior through the environment monitor.

Deter the Ransomware Attacker. Deterring ransomware attackers can be approached in two ways. First, if an attacker realizes he is entrapped, it becomes a deterrent. Second, if the attacker is exposed to the deception environment and remains within the perspective of the defense surveillance, the monitor can collect the attacker's traceable clues that are accidentally released by the attacker (e.g., IP address, path, nickname, strings). The exposure of these clues, hidden from attackers, can be a powerful deterrent to other attackers.

3. Methodology

In this section, we describe our method of tracing back RDP-based ransomware attackers. Figure 1 summarizes the data collection and analysis process of the entire prototype. First, we implement a deception environment to trap attackers. Second, we monitor RDP-based ransomware attacks and collect information when they occur. Third, we extract effective clues from the monitor information. Fourth, we use automatic analysis to screen a large number of clues for tracing back the attacker. Finally, we will generate a report to traceback the RDP-based ransomware attacker. We refer readers to Sections 4 and 5 for the detailed implementation of this prototype.

3.1. Deception Environment. Generally, the ransomware attack execution stage has two steps: login and spread [31]. To build a deception environment is nontrivial in practice because it must make the ransomware attacker believe that it belongs to a real user and the user data is worthy to attack. Because advanced attackers always exploit static features based on certain analysis systems before they launch attacks [32], an intuitive approach to address such reconnaissance attacks is to build the user environment in such a way that the user data is valid, real, and nondeterministic. In addition, the environment serves as an "enticing target" to encourage ransomware attackers. We elaborate on how to generates an artificial, realistic, and enticing user environment for the RDP-based ransomware in Section 4.

The RDP-based attackers commonly upload malicious programs in the following ways before spread ransomware: (1) The attacker downloads malicious programs on the Internet; (2) programs are transferred through FTP, SCP, or other transport protocols; (3) programs are uploaded through the clipboard; (4) programs are uploaded through a shared folder. The clipboard and shared folders are most commonly used to transfer programs by RDP-based ransomware attacks because they are simple and convenient. However, both are easy to monitor by our proposed system.

3.2. Environment Monitor. In order to avoid the attacker's observation and collect more attacker's information, a shared folder and clipboard on the remote PC are always used to transfer ransomware programs from the attacker machine after the attacker logs in to the environment. This paper proposes three monitor layers: the network layer, the host layer and the file layer. We elaborate on how to configure the monitor system for the deception environment in Section 4.

The Network Layer Monitor. The network layer monitor detects a remote connection and collects information including the remote IP addresses, remote ports, status codes of ports, keyboard layout and so on. When the RDP-based attacker logs in to the host, the monitor can obtain information and detect the attack without the attacker's knowledge.

The Host Layer Monitor. We propose to detect changes such as processes and clipboards, by monitoring the host layer. The host layer monitor can gather information about the attacker's behavior and their use of these system applications in the deception environment. For instance, as the clipboard is in the system-level heap space, any application in the system has access to it. The RDP-based ransomware always takes advantage of the clipboard to interact between applications. Moreover, it might get the clues left by the attackers using the clipboard locally, as the Windows system shares the clipboard by default during the RDP session.

The File Layer Monitor. By monitoring the file layer, we can identify ransomware attacks by file changes. Furthermore, it can gather local traceable information by monitoring files in the shared folder. For instance, as a shared folder on the remote PC is always used to transfer ransomware from the attacker machine during the RDP session. In addition, for a more convenient and quick attack, the attacker often mounts the entire local disk to the remote computer. As a result, through the monitor of the shared folder, we can detect the newly-added shared folders in real time and capture a large amount of path information automatically.

3.3. Clue Extraction. Through environmental monitors, it can gather a lot of information left by attackers, such as login information, communication information, clipboard content, folder path, and portable execution (PE) file Many traceable clues can be extracted here, including but not restricted to IP address, keyboard layout, compile path, and file path. In order to analyze these clues quickly, we divided them into two categories: string clues and path clues. These clues are then submitted to the automatic analysis system. We will elaborate the types of clues that the proposed system can extract in Section 5.1.

3.4. Automatic Analysis. According to our investigation, current traceback tools mostly analyze clues manually. However, we usually have to deal with a large amount clues with no semantic correlation. Because such manual traceback analysis usually takes a lot time and efforts, we propose an automatic analysis system and we will elaborate on how to analyze clues automatically in Section 5.2.

4. Implementation of the Deception Environment

As the Windows platform is the main target of ransomware, we choose Windows as the proof of concept implementation. In this section, we describe the implementation details of a Windows-based deception environment prototype. It elaborates on how the deception environment traps ransomware attackers, how the monitor detects the RDP-based attack and collects traceable clues. The entire system implementation process is shown in Figure 2.

4.1. At the Network Layer

4.1.1. The Login Monitor. The login monitor is used to detect attacks in real time and collect the attacker's login information. On the Windows platform, Win32 is an environment subsystem that provides an API for operating system services and functions to control all user inputs and outputs. The login monitor relies on Windows APIs to gain access to the system and run with privileges to access their own areas of memory. It uses Winsock 2.0 to get access to networks and uses protocols other than the TCP/IP suite. The login monitor takes network requests and sends those requests to the Winsock 2.0 SPI (Service Provider Interface) by calling the main Winsock 2.0 file Ws2_32.dll. It provides access to transport service providers and namespace providers. The

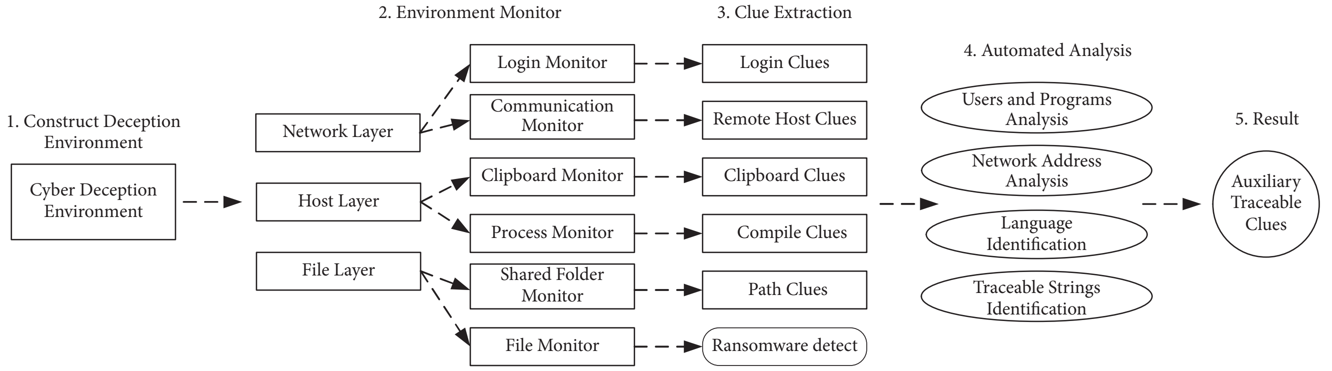


Figure 2: RDP-based ransomware attack traceback system process.

IP Helper API makes it possible to get and modify network configuration settings for the localhost. It consists of the DLL file iphlpapi.dll and includes functions that can retrieve information about the protocols, such as TCP and UDP [33]. As a result, the login monitor can directly access the data buffers involved in transmission control protocols, routing tables, network interfaces, and network protocol statistics.

4.1.2. The Communication Monitor. The communication monitor is responsible for captures network traffic. By locating the TCP packets in the network traffic which contain the interactive configuration information of a RDP login, the RDP connection information can be obtained as the attacker's personal information. The main packet characteristics: (1) The packet's name is often ClientData. (2) The packet location is usually after the TCP three-way handshake. (3) The data packet position in the front. (4) The amount of data is significantly larger.

4.2. At Host Layer

4.2.1. The Deception Host. According to our observation, the main methods used by attackers to login a remote host are: (1) weak password direct login; (2) add access account login through vulnerability. As a result, we deliberately set the administrator privileges of the deception environment as weak passwords and leave common vulnerabilities in the environment, such as *EternalBlue*, to attract attackers. To guide attackers to upload ransomware using only clipboard and shared folders, we block external traffic transfers from outside the environment and close common transfer ports (e.g., port 20, 21, 80, and 443).

4.2.2. The Clipboard Monitor. The clipboard monitor can obtain clues in real time by monitoring the clipboard's changes. It uses Clipboard Viewer to listen to message changes in the clipboard without affecting its contents. The Clipboard Viewer is a mechanism that can get and display the contents of the clipboard. As Windows applications are message-driven, the key to the monitor is responding to and processing clipboard change messages. When the content changes, the monitor triggers the *WM_DRAWCLIPBOARD* message and sends the changed message to the first window of the Clipboard Viewer Chain. After each Clipboard Viewer window responds to and processes the message, it must send the message to the next window according to the handle of the next window in its saved linked list. The clipboard monitor can obtain the clipboard's new contents by using the "*GetClipboardData*" Windows API through the window. When a subsequent copy or cut operation are executed, the data in the clipboard are rewritten. As a result, the clipboard monitor guarantees real-time listening and writes real-time information to the log file. The log file is updated whenever the clipboard monitor receives a clipboard change notice. When the log file is updated, to prevent it from being detected by the attacker or being encrypted by ransomware, the monitor sends it to a secure host and completely erases it from the environment.

4.2.3. The Process Monitor. To run a program on a Windows system, a new process must be created. The monitor gets the PE file run by the attacker by monitoring the environment process. It first records the state of processes commonly used in the deception environments before a RDP-based ransomware attack. After the login monitor detects such an attack, it monitors the system for newly created processes in the environment through the Windows API. "*CreateToolhelp32Snapshot*" takes status snapshots of all processes in real time, which includes the process identifier (PID). When a suspicious process has started, the monitor recognizes it by the PID and looks up the process's running path with the help of "*GetModuleFileNameEx*". Finally, the monitor finds the suspicious program's PE file through the path and copies it to the secure host.

4.3. At the File Layer

4.3.1. Deception Files. Deception files are constructed with two goals. First, we need to make the attacker believe that the deception environment is a real user's host. Second, we need to make the attacker believe that there are resources in the environment that are worthy of attacking.

To simulate a realistic environment, we deploy large numbers of different types of files on it, for example, images, audio files, database files, and documents that can be accessed in a Windows session. Based on *Amin Kharraz's* research [1], we created four file categories that ransomware always tries to find and encrypt: documents (∗.txt, ∗.doc(x), ∗.ppt(x), ∗.xls(x), ∗.pdf and ∗.py), keys and licenses (∗.key, ∗.pem, ∗.crt, and ∗.cer), file archives (∗.zip, ∗.rar), and media (∗.jp(e)g, ∗.mp3, and ∗.avi). We obtain these files in three ways. First, we create files with valid headers and content using standard libraries (e.g., python-docx, python-pptx, pdfkit, and OpenSSL). Second, using Google search syntax and crawler technology, we download a large number of files on the Internet. Third, we collect a number of non-confidential documents from the hosts of 20 volunteers to emulate actual user environments. When we assign user files for the deception environment, the path length is generated randomly. Each folder may have a set of subfolders randomly. For each folder, a subset of extensions is randomly selected. Furthermore, each directory name is generated based on meaningful words. Consequently, we generate paths and extensions for user files, giving them variable file depth and meaningful content.

To make the simulated environment more valuable, we deploy bait information on it, such as database, false code comments, digital certificates, administrator password, SSH keys, VPN keys, browser history passwords, ARP records, and DNS records. When the bait information is obtained by an attacker, it may trick it to attack the deception environment.

4.3.2. The File Monitor. The file monitor can detect the ransomware by monitoring file type changes and file entropy changes, which method is proposed in 2016 [21]. The type of data stored in a file can describe the order and position of specific byte values unique to a file type. Since files generally retain their file types and formatting in the deception environment, the bulk modification of such files should be considered suspicious. When the monitor sees this type of changes, we can infer that a ransomware attack has occurred.

Entropy can express the randomness of each character in a string. The higher the entropy value, the stronger the randomness. The Shannon entropy of an array of bytes can be computed as the sum:

$$e = \sum_{i=0}^{255} P_{B_i} \log_2 \frac{1}{P_{B_i}} \tag{1}$$

for $P_{B_i} = F_i/totalbytes$ and F_i, the number of instances of byte value i in the array. As the entropy value is represented by a number from 0 to 8, the entropy value of 8 represents the byte array composition of its completely uniform distribution. Since the probability of each byte occurring in the encrypted ciphertext is basically the same, the entropy value will be close to the upper limit. Because the ransomware always encrypts a large number of files, when we detect that a file change to a high entropy value file in a short period of time and also change the file type, we assume that the file is subject to a ransomware attack.

4.3.3. The Shared Folder Monitor. By traversing the disk storage in real time, the shared folder monitor discovers the updates of the shared folders in real time. It obtains the contents of the attacker's files locally, which are often not noticed by the attacker, thus revealing some unexpected traceable clues. The monitor can access a list of paths to the attacker shared folders.

As we originally observed, shared folders using Remote Desktop often have a path in the remote host with the prefix “\\tsclient\”. When the monitor traverses the storage to this prefix, it uses “*FindFirstFile*” to find the first file. It then uses “*FindNextFile*” to find the next file with the returned handle. When the resulting handle is in a folder format, it continues to traverse all files under that folder. Initially, the monitor tries to get the full file names and file contents by traversing the new shared folder. However, during the actual experimentation, it is found that as the number of files in storage grows, the monitor takes far more time and resources to get all the file contents than just the file paths. All traverses are more likely to alert the attacker. Therefore, the monitor only obtains the file paths that the attacker shares on its host with the help of “*GetFileName*”. Moreover, in order to prevent encryption by the ransomware, the mounted disk monitor will directly transfer the acquired shared file path list to another secure host.

5. Clue Extraction and Analysis

Through the deception environment, we trap the ransomware attacker and collect a lot of information that may contain many traceable clues. However, such traceable clues are often not visually observable and are complex in nature. In addition, many of the above clues contain information that is not helpful in tracing back ransomware attackers. Therefore, in order to assist in the analysis of the monitor information and extract the effective main traceable clues, in this section we propose how to extract clues and how to analyze traceable clues using an automatic approach after extraction.

5.1. Clue Extraction. We mainly obtain kinds of clues from the extraction, including remote login information (IP addresses), network traffic, clipboard contents (pictures and texts), shared folder information (path strings), and ransomware samples (compile time and compile paths). The shared folder path clues can be obtained directly from the monitor. However, clipboard clues, compile clues, and remote host clues are often not visually observable and are complex in nature. As a result, the extraction module mainly focuses on the extraction of remote host clues, clipboard string clues, and compilation clues.

Remote Host Clue Extraction. The IP address, port numbers, and folder path clues can be directly obtained from the login information and folder paths. TCP packets that interact with the configuration information in a network communication PCAP package are usually named “*ClientData*”. We extract the client name field from the “*ClientData*” packet to obtain the attacker’s hostname. In addition, the *KeyboardLayout* field indicates the default keyboard layout for an attacking host; e.g., the Chinese Simplified layout number is 0x0004, and the American English keyboard layout number is 0x0409. The remote user's idiomatic language (the mother tongue) can be found by the keyboard layout to infer the attacker's nationality.

Clipboard Clue Extraction. The main file formats available to the clipboard monitor are Windows Bitmap, GDI file, ANSI characters, Unicode characters, and WAV audio data. We mainly aim at extracting the traceable clues of character types. It extracts character clues from the clipboard in various formats by judging the *GetClipboardData* API's "*DataType*" value.

Compilation Clue Extraction. For all Windows RDP-based ransomware samples that we examined, we empirically observed that the most commonly used formats for these samples are the PE file, especially ∗.exe and ∗.dll; some PE files have compilation information in the file, and this information does not change with the migration of the programs. As a result, it is a good way to obtain the creator's information. A PE file mainly consists of five major components: DOS MZ header, DOS stub, PE header, section headers, and section content. Each component contains a great deal of information. There is very little information that we can use to identify the creator, and some identification information needs to be extracted from the content of each section. In this paper, there are many clues in the PE files that can be extracted to trace back the attacker: file name, PE file type, compiler version, compilation path, compile time, last modified time, last open time, IP address, URL, domain name, language, string, wide character, and so on. We extract most of the clues with *PEView* [34]. However, since it cannot directly obtain the compilation path, we use the *pefile* [35] tool to extract paths by locating in the PE structure.

Since the extraction clues include different encoding formats, to facilitate observation and the unified mode of subsequent analysis, the extraction system completely converts the data encoding obtained into the UTF-8 format and saved in the SQLite database. Before submitted to the analysis system, the extracted clued are divided into two categories: string clues and path clues.

5.2. Automatic Analysis. At this point, the clues extracted from this system mainly include string clues and path clues. The number of string clues is large, mixed with a large number of unidentifiable strings. Because the number of path clues is also very large with no semantic correlation, it is difficult to identify traceable clues manually. So we focus on how to automate the identification of traceable clues for path clues.

5.2.1. Users and Programs Analysis. In the analysis of the path clues, we first propose to obtain the attacker clues by identifying the features of the context-related segmentation on the same path. For instance, each user has a separate user folder, and it is located in the "Users" folder under Drive C. As a result, the system can obtain the user name at the attacking host by obtaining the folder name under the "Users" folder (e.g., *C:\Users\Dell\...*). The "Program files" folder usually contains the name of the software program installed on the machine (e.g., *C:\Program Files\Microsoft Visual Studio 11.0\...*). What is more, the QQ account number is always located in the "*\QQ\QQfile\*" folder (e.g., *D:\QQ\QQfile\86∗ ∗ ∗ ∗ ∗∗086\FileRecv\...*).

In this way, the analysis system can quickly and accurately get host names, email accounts, program names, social software numbers, and other traceable clues that are carried in the attacker's file paths. However, such user information for less of the overall clues is acquired by chance. Therefore, we will conduct further analysis on this basis.

5.2.2. Account Analysis. Through the analysis of *APT1* and some other attribution reports, we find that the mapping between the attacker and the physical world identity can be better obtained by analyzing the account number left by the attacker. This information includes but is not limited to the location of the IP address, the spelling and registration of the domain name, the URL corresponding IP address, and the domain name of the mailbox account. Because it is difficult to identify this information effectively in a large number of strings and path clues, the analysis system automatically identifies the IP address, domain name, URL, and mailbox account by regular matching. Then, with the help of threat intelligence and big data technology, more relevant clues are obtained.

5.2.3. Language Identification. User languages often help to determine an attacker's idiomatic language, but because of a large number of languages in different countries and the high similarity of some languages, we use automatic analysis systems to identify the language of clues. We test the accuracy of two language identification toolkits using entire path information in four different languages. We have found "*langid.py*" toolkit to be overall more accurate than "*langdetect*" toolkit. The comparison results are shown in Figure 3. The *langid.py* is a language identification toolkit developed by *Lui* and *Baldwin* at the University of Melbourne [36]. It combines a naive Bayes classifier with cross-domain feature selection to provide domain-independent language identification.

5.2.4. Traceable Strings Identification. When traceable strings are needed, traditional string analysis methods usually use Named Entity Recognition (NER). However, the clues to be analyzed mainly include strings and path. Strings before and after the path separator have few semantic correlations. What is more, the string between the path separators and the remaining strings to be analyzed are mostly semantically unrelated due to their limited length. So this paper proposes an algorithm, which can quickly and automatically analyze the traceable strings in strings and path.

In order to filter out meaningful traceable clues related to the attacker's identity, the path clues and string clues are split into strings and identified by common words and gibberish in the following steps. Figure 4 shows the automatic traceable clues identification system process.

Make Stop Words. The system splits the path by the path delimiter, as these separated path strings that are common to multiple computers have no identifying effect. So, we take out the file string names that are common to 20 normal user

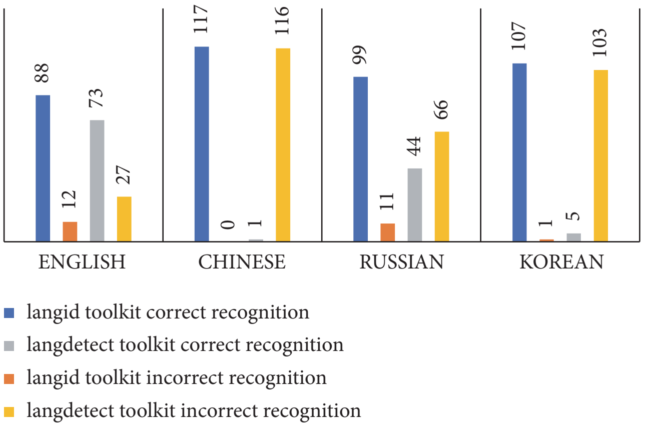


Figure 3: Accuracy of Two Language Identification Tools: In tests using the same known language path clues, the accuracy of the *langid* toolkit for English, Chinese, Russian and Korean was 88%, 100%, 99%, 99.0% respectively, while the accuracy of the langdet toolkit was 73%, 0.85%, 40%, and 4.60%, respectively.

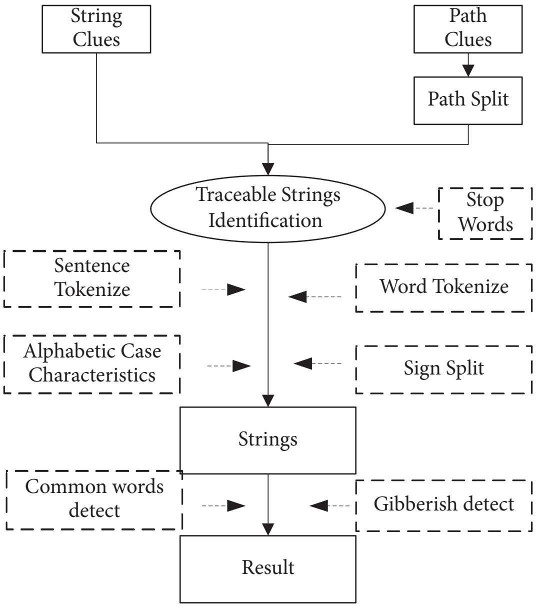


Figure 4: Process of automatic identifying of traceable clues.

computers as stop words. Then, it removes the stop words from the strings after each split.

Segmentation. After removing the stop words, we separate sentences and words for each segment of the path with tokenize (e.g., NLTK, NER) [37]. However, as the segments of the path are not semantically related, the segmentation is less effective. Based on our extensive research on path information, users often prefer to use symbols such as underscores to split file names. In addition, they also like to use capitalization in multiword strings to facilitate reading. As a result, the system uses the common identifier in the document and alphabetic case characteristic to segment the strings again.

Common Words Removal. After segmentation, the system obtains a lot of repeatable strings. But most of them are commonly used words and does not help in identifying ransomware attackers. As a result, we filter out the common words by comparing each string with a dictionary. If the string can be matched in a dictionary, the system marks the common word property of the word as false, otherwise, as true.

Gibberish Detect. Based on our analysis and observation, we found that there is a large amount of garbled information in path clues. Most of these strings are generated randomly, and people can recognize this randomness manually, but it is difficult for a program to recognize it automatically. As a result, we propose detecting the gibberish by training the Markov chain model with English texts. For each string X, there is a probability that the i character in X is

$$\begin{aligned} \mathrm{P}\left\{X_{i+1} = x \mid X_1 = x_1, X_2 = x_2, \ldots, X_i = x_i\right\} \\ = P\left\{X_{i+1} = x \mid X_i = x_i\right\} \end{aligned} \tag{2}$$

Each letter is related to the upper and lower two letters. This relationship can be expressed by 2-gram. For example, with regard to the string "*Ransomware*": *Ra*, *an*, *ns*,. . .*e[space]*. The analysis system can record how often the characters appear next to each other, through the collection of gibberish and some commonly used normal phrases or vocabulary. It normalizes the counts, after reading through the training data and then measures the probability of generating the string based on the digest by multiplying the probability of the pair of adjacent characters in the string [38]. The training data can help statistics corresponding to gibberish and normal strings, respectively, the average transfer probability. This probability then measures the amount of possibilities assigned to this string according to the data as observed by the model. When we test the string "Ransom," it computes

$$\begin{aligned} prob\left[{}'Ransom'\right] = prob\left[{}'R'\right]\left[{}'a'\right] \times prob\left[{}'a'\right]\left[{}'n'\right] \\ \times \cdots \times prob\left[{}'m'\right]\left[{}''\right] \end{aligned} \tag{3}$$

If the input string is gibberish, it will pass through some pairs with very low counts in the training phase and hence have low probability. The system then looks at the amount of probability per character for a few known normal strings and a few examples of known gibberish and then picks a threshold between the gibberish's most possibilities and the normal string's least possibilities:

$$\begin{aligned} & threshold \\ & \quad = \frac{\left(min\left(nomalprobs\right) + max\left(gibberishprobs\right)\right)}{2} \end{aligned} \tag{4}$$

When we analyze the string 'X', if $prob['X'] > threshold$, the analysis system will view the string as normal string. If $prob['X'] \le threshold$, the analysis system will view the string as 'False.' Then, the system removes the strings with the 'False' type as gibberish.

Table 1: Traceable strings recognized result [13].

String	Common Words Recognized	Gibberish Recognized	Final Recognized
Wh* * **ter	True	True	True
gandcrab	True	True	True
kate_z	True	True	True
vwrtjty	True	False	False
program	False	True	False

Identification Result. After the above analysis, the automated analysis system outputs the strings for which both the common word tag and the gibberish tag are True, which is the list of traceable clues [13]. Table 1 is reproduced from Z. H. Wang et al. (2018) [under the Creative Commons Attribution License/public domain].

When the string is a normal word, such as "program", the common words recognition will make it "False." Gibberish recognition can recognize the word only with non-randomly generated identifier strings. Moreover, it cannot identify the words that do not conform to the spelling patterns of common words (e.g., vwrtjty). The string is considered to have auxiliary traceability only if both of the analysis values are true. We rely extensively on string inversion to verify the accuracy of the system. It is found that most of the traceable strings on the volunteer storage can be recognized. Table 1 shows the recognition results of several typical strings.

6. Evaluation

In this paper, we evaluate the proposed method with two experiments. The goal of the first experiment is to demonstrate that the proposed method can help trace back to the RDP-based ransomware attackers and capture their private information. The goal of the second experiment is to demonstrate that the method can also automatically recognize clues in ransomware samples and help trace back to ransomware makers.

6.1. RDP-Based Ransomware Attacker

6.1.1. Clue Capture and Analysis. We used Windows 7 system virtual machines to deploy the deception environment and assess the effectiveness of it. While Windows 7 is not required, it was chosen because of the wide range of applications and because it is one of the main targets of ransomware. We invited 122 professional volunteers to help with the experiment and provided them with an experimental fleet of 12 virtual hosts. Most of the volunteers' login information, clipboard content, shared folder path, and uploaded PE file were able to be successfully captured by the monitor system.

We choose 12 computers' path the information collected from volunteers and record the rate of convergence after each step of the analysis. Figure 5 shows how the number of traceable clues remains as each step of the data is processed by the analysis system [13]. Figure 5 is reproduced from Z. H. Wang et al. (2018) [under the Creative Commons Attribution License/public domain].

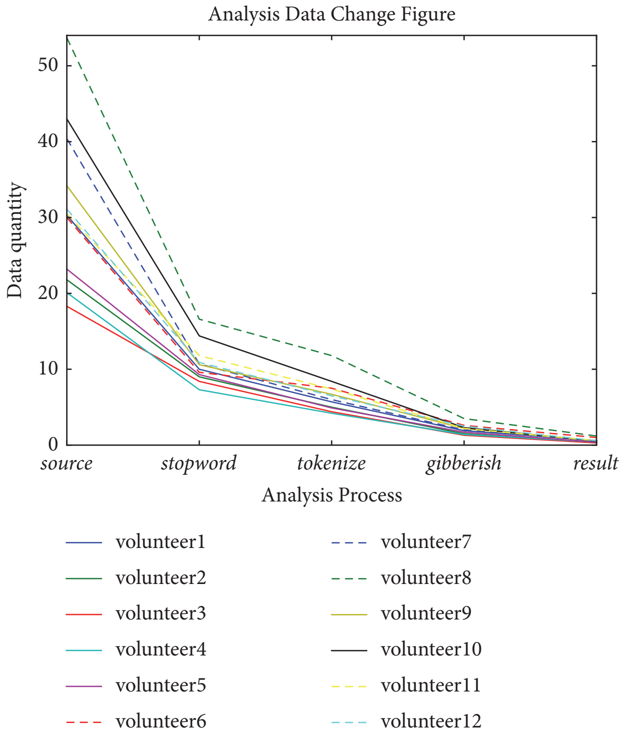


Figure 5: The data volume changes in 12 volunteers' traceable clues through the analysis system. The screening rate can reach 98.34% [13].

6.2. Traceback Attackers. Next, we carried out a detailed analysis of the information left over from one of the volunteers' attacks. By understanding the data from the automatic analysis of an attacker, we can infer that the attacker's real identity may have the following characteristics: (1) The attacker is likely to come from China. (2) The attacker may be a security and related personnel. (3) The attacker may work in the Chinese Academy of Sciences and have relations with a large security manufacturer in China. The conclusion is shown in Figure 6. It shows the results from the analysis system in five sections.

Username. Through automatic analysis, a total of three user names from the attacker's host were extracted from a large number of clues, in which, by using the "Dell" user, it can be assumed that the attacker was using the Dell host.

Programs. The automatic analysis system identified several typical software programs installed in the attacker's host. For instance, "QQ" is a widely used real-time social tool in China; "AliPay" is an online payment tool developed by Alibaba and widely used in China; "Sogou input" is an input method software for Chinese developed by China Sogou Company; MeiTu is a photo beautification software developed by a Chinese company and widely used in China; In addition, "Sinfor" and "360" are both well-known security manufacturers in China and have a large number of safety-related products; "Visual Studio" is a common development software. It is found that much Chinese-made software widely

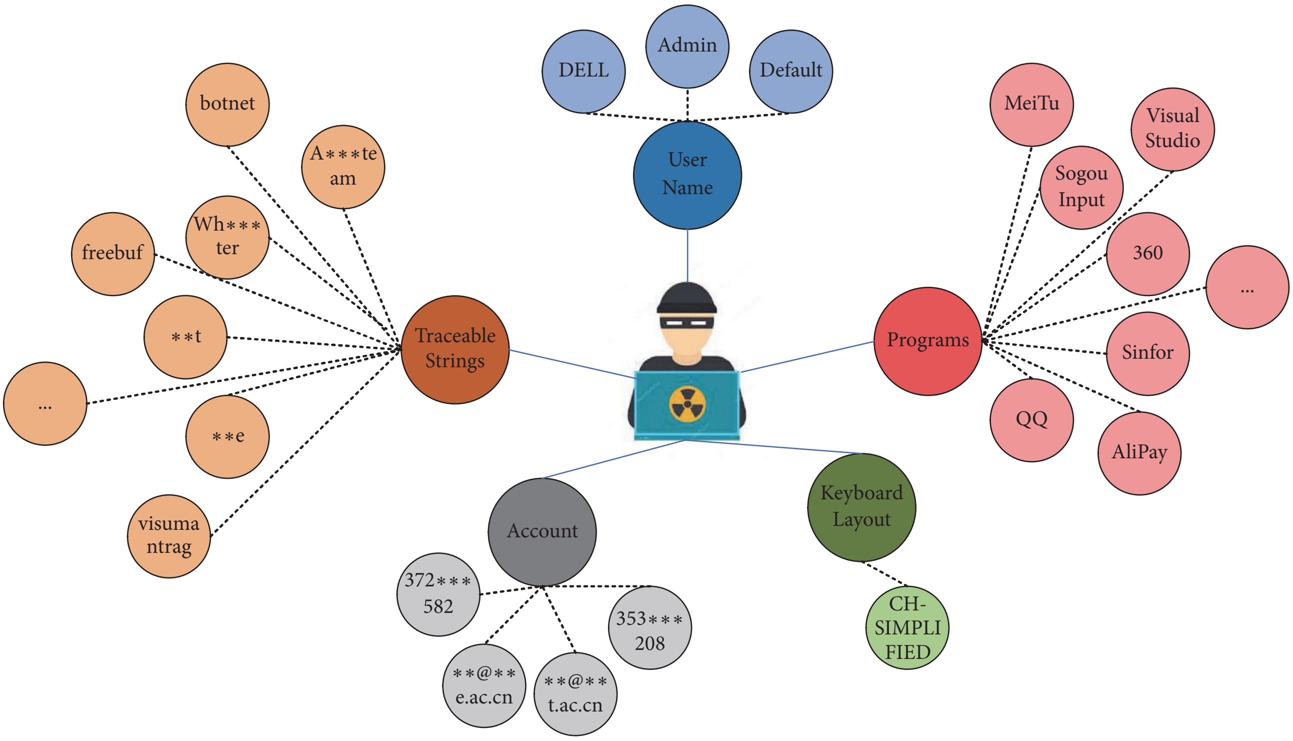
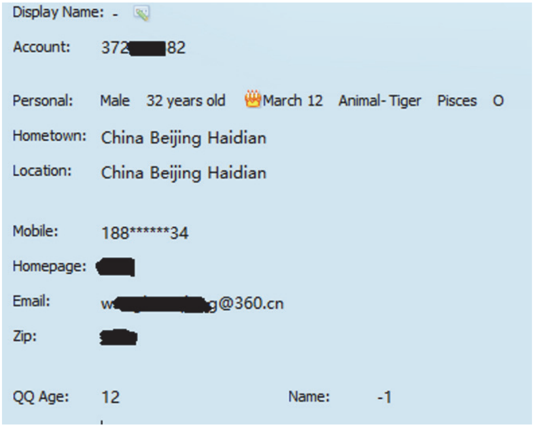


Figure 6: Analyze an attacker using automatic analysis system to display traceable information about the attacker in five aspects: host user name, program software, keyboard layout, account information, and traceable string.

used in China was installed on the attackers' hosts, as well as some development and security products that were installed less often by ordinary users.

Keyboard Layout. Through network communication analysis, we obtain the attacker's keyboard layout is Simplified Chinese, from which it can be inferred that the attacker's native language is probably Chinese. This is consistent with the host installed software features.

Account Number. According to the system automatically identified account number, it found two QQ accounts and two e-mail accounts. The e-mail accounts for the Chinese Academy of Sciences business accounts can confirm the above assumption that the attackers from China, and it is likely to work in the Chinese Academy of Sciences. Through the registration information of the QQ account (as shown in Figure 7), we can find the following information: (1) the attacker could be a male, aged 32; (2) Internet-related work, likely with *360 Security*; (3) located in *Haidian District, Beijing, China*. In the "353∗ ∗ ∗208" account registration information, mail, work, and other information are empty, nicknamed a special English string: "*Wh∗ ∗ ∗ter*" it can be presumed that the account is likely to be a private use.



Display Name: -

Account: 372[redacted]82

Personal: Male 32 years old March 12 Animal- Tiger Pisces O

Hometown: China Beijing Haidian

Location: China Beijing Haidian

Mobile: 188******34

Homepage: [redacted]

Email: w[redacted]g@360.cn

Zip: [redacted]

QQ Age: 12 Name: -1

Figure 7: The registration infographic of QQ account 372∗ ∗ ∗82.

Traceable Strings. The automatic analysis system sifts strings from the monitor that may be traceable. "*Freebuf*" is a security information exchange website commonly used by Chinese security personnel. "*Bootnet*" is a secondary attack method that is often used by attackers, and is often used by security researchers as the main research direction. "∗∗*e*" and "∗∗*t*" are acronyms for departments under the Chinese Academy of Sciences, while "*A∗ ∗ ∗team*" is a security research team in the "∗∗*e*" department. "*Visumantrag*" is often used when processing visas from various countries. The "*Wh∗ ∗ ∗ter*" matches the nickname of the QQ number "*353∗ ∗ ∗208*" account and is likely to be its regular nickname.

6.3. Ransomware Maker Automatic Analysis. In order to verify that the analysis system is available in tracing back the ransomware maker, we obtained more than 8000 ransomware

Table 2: Same identifier for different samples [13].

Sample Hash	Same Identity Strings
68dd613973f8a∗ ∗ ∗7befe0f4b461d258ce569b9020079∗ ∗ ∗9ae0f090266a60cc	jqxxppho/aonk/je7z2
fa0c084aef41969∗ ∗ ∗1f427a1b65e1b1464fa82d297e712∗ ∗ ∗e1fdf312c04481	
4a36da0286d42cd∗ ∗ ∗306fba99239c8238821beaf40744be∗ ∗ ∗ba862dbc6d90f	
70cbc839bcb88∗ ∗ ∗7422bbd2fa812dd3de02391b6a9f1∗ ∗ ∗0762a6f49cf32501	

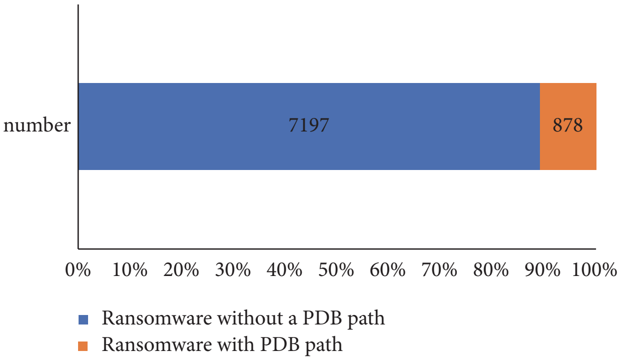


Figure 8: Of the 8,075 real ransomware samples, 11% were able to extract the Program Data Base (PDB) path, while the remaining 89% of authors may have chosen to hide the PDB path during compilation.

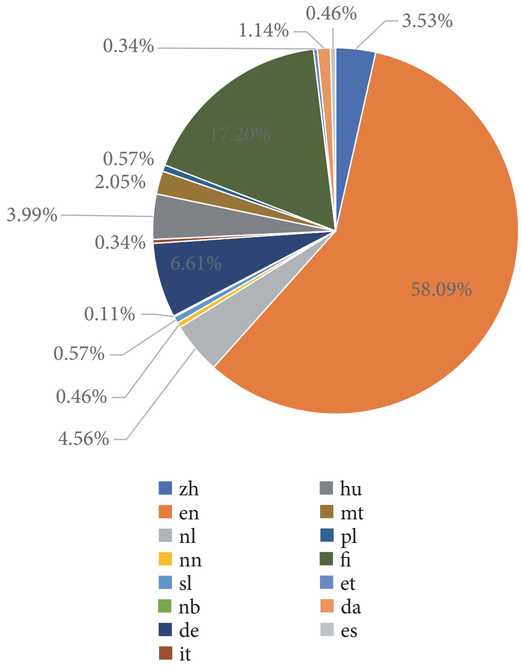


Figure 9: Of the 878 ransomware sample PDB paths, English accounted for 510, and the remaining monitored languages were Chinese: 31(3.53%), Dutch: 40(4.56%), Norwegian Nynorsk: 4(0.46%), Slovenian: 5(0.57%), Norwegian Bokmål: 1(0.11%), German: 58(6.61%), Italian: 3(0.34%) Hungarian: 35(3.99%), Maltese: 18(2.05%), Polish: 5(0.57%), Finnish: 151(17.20%) Luxembourgish: 3(0.34%), Danish: 10(1.14%), and Spanish: 4(0.46%).

samples from *VirusTotal* [39] and extracted the Program Data Base (PDB) path from the samples (as shown in Figure 8). It is shows that, despite the large number of attackers' deliberate erasure of the compiled information, legacy PDB path information can still be found from a large number of ransomware samples.

What is more, the analysis system is used to automatically analyze more than 800 different ransomware samples' PDB path. We found multiple identity strings in the analysis results of different samples. One of the typical results is shown in Table 2 [13]. Table 2 is reproduced from Z. H. Wang et al. (2018), [under the Creative Commons Attribution License/public domain]. We used *VirusTotal* to validate the samples and found that they were all Cobra family ransomware. As a result, it could be assumed that these samples are made by the same ransomware maker. When one sample is traced back, other samples can also arrive at the conclusion of the ransomware maker. When different traceable information is analyzed from different samples, the identity information of the ransomware maker can be described by integrating the information.

Language Identification. When we carry on the language recognition to the PDB path of 878 samples, the result is as Figure 9. The system automatically identifies the path which mainly contains 11 languages, of which English accounts for the largest proportion.

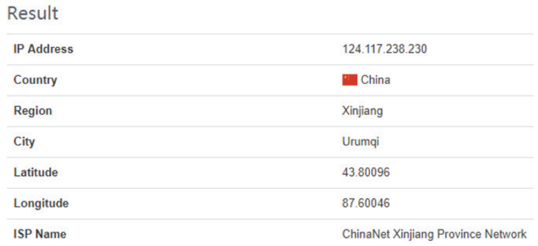
Actual Case Analysis. The automatic identification system helped us identify the traceable strings in the PDB path. Take the example of a sample PDB path (the Chinese has been translated into English) with an MD5 value of "60c6a92afb∗ ∗ ∗6d0c16f7".

"*E:\\LIAO\\STUDIO\\UAC\\simulated an improved version of the 360 anti-virus program 1117 Bale 1∗4.1∗∗.2∗∗.∗∗0(∗∗1)_9818 program*\\WriteSystem32\\Release\\VirtualDesktop.pdb"

We can infer from the automatic analysis result that the ransomware maker is likely from China. We can prove this point by the following:

(1) Language Identification: This is done by identifying the path language, which can be used to speculate that the attacker may have come from China. By understanding the Chinese strings in the path, we find that the meaning of this text is to improve to bypass the detection of 360, security software that has a large market in China.

(2) IP address: The system identified an IP address in the PDB path. With the help of threat intelligence database, it is found that the IP came from *Xinjiang, China* (Figure 10).



Result

IP Address	124.117.238.230
Country	🟥 China
Region	Xinjiang
City	Urumqi
Latitude	43.80096
Longitude	87.60046
ISP Name	ChinaNet Xinjiang Province Network

Figure 10: Partial infographic of IP: 1∗4.1∗∗.2∗∗.∗∗0 in the threat intelligence library.

(3) Traceable Strings: The extracted identifier string "*LIAO*" resembles a Chinese pinyin and is most likely a Chinese surname. To sum up, we speculate that the ransomware maker is most likely from China.

7. Conclusion

In this paper, we focus on tracing back RDP-based ransomware. Recently, more and more ransomware attackers are using RDP attacks to spread ransomware with impunity due to the use of strong cryptography. We propose tracing back the attack sources to deter RDP-based ransomware with the proposed deception environment. It collects traceable clues and performs automatic analysis by using natural language processing and machine learning techniques. The evaluation shows that it is able to trap attackers and collect traceable clues left by attackers in a deception environment. With automatic clue identification, it can converge the amount of traceable clues to about 2%. We use this method to analyze two practical cases and show its effectiveness. By tracing back to ransomware attackers, we can provide a strong deterrent to stifle the development of ransomware.

Data Availability

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

Additional Points

This manuscript provides additional deception environment monitors and analytical methods by using more volunteers and samples. It proves the validity of the methods through specific traceback cases (i.e., traceback the ransomware attacker and traceback the ransomware maker).

Disclosure

An earlier version of this paper was presented at the International Conference: IEEE Third International Conference on Data Science in Cyberspace, Guangzhou, China, 18-21 June 2018. The authors' initial conference paper focused mainly on the effectiveness of deception environment monitors and the screening rate of the analysis system.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank VirusTotal for providing ransomware samples and the volunteers from the University of Chinese Academy of Sciences. This work is supported in part by the National Key research and Development Plan (Grant no. 2018YFB0803504) and the National Natural Science Foundation of China (61871140, 61572153, U1636215, 61572492, and 61672020).

References

[1] A. Kharraz, "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," in *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, pp. 757–772, USENIX Association, 2016.

[2] K. Savage, P. Coogan, and H. Lau, "The evolution of ransomware," Tech. Rep., 2015.

[3] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 9148 of *Lecture Notes in Computer Science*, pp. 3–24, Springer International Publishing, Cham, 2015.

[4] S. Mohurle and M. Patil, "A brief study of Wannacry Threat: Ransomware Attack 2017," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, 2017.

[5] X. Yu, Z. Tian, J. Qiu, and F. Jiang, "A Data Leakage Prevention Method Based on the Reduction of Confidential and Context Terms for Smart Mobile Devices," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5823439, 11 pages, 2018.

[6] J. Yaneza, "Brute Force RDP Attacks Plant CRYSIS Ransomware," https://blog.trendmicro.com/trendlabs-security-intelligence/brute-force-rdp-attacks-plant-crysis-ransomware/.

[7] "10% of Ransomware Attacks on SMBs Targeted IoT Devices," https://www.darkreading.com/application-security/10–of-ransomware-attacks-on-smbs-targeted-iot-devices-/d/d-id/1329817.

[8] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. H. Tian, "Towards a Comprehensive Insight into the Eclipse Attacks of Tor Hidden Services," *IEEE Internet of Things Journal*, 2018.

[9] "Kaspersky Security Bulletin: STORY OF THE YEAR 2017," 2017, https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07164824/KSB_Story_of_the_Year_Ransomware_FINAL_eng.pdf.

[10] Z. Tian, Y. Cui, L. An et al., "A Real-Time Correlation of Host-Level Events in Cyber Range Service for Smart Campus," *IEEE Access*, vol. 6, pp. 35355–35364, 2018.

[11] R. Ross et al., *Managing Information Security Risk: Organisation, Mission, and Information System View*, National Institute of Standards and Technology, 2011, http://csrc.nist.gov/publications/nistpubs/800-39/SP800-39-final.pdf.

[12] Y. Wang, Z. Tian, H. Zhang, S. Su, and W. Shi, "A Privacy Preserving Scheme for Nearest Neighbor Query," *Sensors*, vol. 18, no. 8, p. 2440, 2018.

[13] Z. H. Wang, X. Wu, C. G. Liu, Q. X. Liu, and J. L. Zhang, "RansomTracer: Exploiting Cyber Deception for Ransomware Tracing," in *Proceedings of the IEEE Third International Conference on Data Science in Cyberspace*, pp. 227–234, 2018.

[14] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A

survey and research directions," *Computers & Security*, vol. 74, pp. 144–166, 2018.

[15] F. Jiang, Y. Fu, B. B. Gupta et al., "Deep Learning based Multichannel intelligent attack detection for Data Security," *IEEE Transactions on Sustainable Computing*, 2018.

[16] N. Andronio, S. Zanero, and F. Maggi, "HelDroid: dissecting and detecting mobile ransomware," in *Research in Attacks, Intrusions, and Defenses*, vol. 9404 of *Lecture Notes in Computer Science*, pp. 382–404, Springer, 2015.

[17] F. Mercaldo, V. Nardone, A. Santone, and C. A. Visaggio, "Ransomware Steals Your Phone. Formal Methods Rescue It," in *Formal Techniques For Distributed Objects, Components, And Systems: 36th IFIPWG 6.1 International Conference, FORTE 2016, held as part of the 11th International Federated Conference On Distributed Computing Techniques, DisCoTec 2016*, E. Albert and I. Lanese, Eds., pp. 212–221, Springer International Publishing, 2016.

[18] D. Sgandurra, L. Mu±oz-Gonzßlez, R. Mohsen, and E. C. Lupu, "Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection," https://arxiv.org/abs/1609.03020.

[19] S. Song, B. Kim, and S. Lee, "The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform," *Mobile Information Systems*, vol. 2016, Article ID 2946735, 9 pages, 2016.

[20] T. Yang, Y. Yang, K. Qian, D. C.-T. Lo, Y. Qian, and L. Tao, "Automated detection and analysis for android ransomware," in *Proceedings of the 17th IEEE International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security and IEEE 12th International Conference on Embedded Software and Systems, HPCC-ICESS-CSS 2015*, pp. 1338–1343, USA, August 2015.

[21] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," in *Proceedings of the 36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016*, pp. 303–312, Japan, June 2016.

[22] M. M. Ahmadian and H. R. Shahriari, "2entFOX: A framework for high survivable ransomwares detection," in *Proceedings of the 13th International ISC Conference on Information Security and Cryptology, ISCISC 2016*, pp. 79–84, Iran, September 2016.

[23] M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, "Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares," in *Proceedings of the 12th International ISC Conference on Information Security and Cryptology, ISCISC 2015*, pp. 79–84, Iran, September 2015.

[24] D. Kim, W. Soh, and S. Kim, "Design of Quantification Model for Prevent of Cryptolocker," *Indian Journal of Science and Technology*, vol. 8, no. 19, 2015.

[25] C. Moore, "Detecting Ransomware with Honeypot Techniques," in *Proceedings of the 2016 Cybersecurity and Cyberforensics Conference (CCC)*, pp. 77–81, Amman, Jordan, August 2016.

[26] F. Mbol, J. Robert, and A. Sadighian, "An Efficient Approach to Detect TorrentLocker Ransomware in Computer Systems," in *Cryptology and Network Security*, S. Foresti and G. Persiano, Eds., vol. 10052 of *Lecture Notes in Computer Science*, pp. 532–541, Springer International Publishing, Cham, 2016.

[27] K. Cabaj, P. Gawkowski, K. Grochowski, and D. Osojca, "Network activity analysis of CryptoWall ransomware," *Przegląd Elektrotechniczny*, vol. 91, no. 11, pp. 201–204, 2015.

[28] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, "Trust Architecture and Reputation Evaluation for Internet of Things," *Journal of Ambient Intelligence & Humanized Computing*, vol. 2, pp. 1–9, 2018.

[29] C. Le Guernic and A. Legay, "Ransomware and the Legacy Crypto API. Paper presented at the Risks and," in *Risks and Security of Internet and Systems: 11th International Conference, CRiSIS 2016*, Roscoff, France, 2017.

[30] L. Pingree, *Emerging Technology Analysis: Deception Techniques and Technologies Create Security Technology Business Opportunities*, Gartner, 2015.

[31] "Ransomware and Businesses," https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/ransomware-and-businesses-16-en.pdf, 2016.

[32] A. Kharraz and E. Kirda, "Redemption: Real-Time Protection Against Ransomware at End-Hosts," in *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 98–119, 2017.

[33] F. Bergstrand, J. Bergstrand, and H. Gunnarsson, Localization of Spyware in Windows Environments,.

[34] "PEView," https://www.aldeid.com/wiki/PEView.

[35] "python-pefile," https://pypi.python.org/pypi/pefile.

[36] M. Lui and T. Baldwin, "langid., py, An off-the-shelf language identification tool," in *Proceedings of the ACL 2012 system demonstrations. Association for Computational Linguistics*, pp. 25–30, 2012.

[37] S. Bird and E. Loper, "NLTK: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics*, Barcelona, Spain, July 2004.

[38] "Rrenaud. Gibberish-Detector," https://github.com/rrenaud/Gibberish-Detector/blob/master/README.rst.

[39] "VirusTotal," http://virustotal.com.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3473910, 13 pages
https://doi.org/10.1155/2018/3473910

Research Article

Enjoy the Benefit of Network Coding: Combat Pollution Attacks in 5G Multihop Networks

Jian Li,[1] **Tongtong Li**,[2] **Jian Ren**,[2] **and Han-Chieh Chao**[3,4]

[1] *School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2] *Department of ECE, Michigan State University, East Lansing, MI 48824-1226, USA*
[3] *Department of Electrical Engineering, National Dong Hwa University, Taiwan*
[4] *Institute of Computer Science & Information Engineering and Department of Electronic Engineering, National Ilan University, Taiwan*

Correspondence should be addressed to Jian Li; lijian@bjtu.edu.cn

Received 27 September 2018; Accepted 15 November 2018; Published 2 December 2018

Guest Editor: Karl Andersson

In the upcoming 5G era, many new types of networks will greatly expand the connectivity of the world such as vehicular ad hoc networks (VANETs), Internet of Things (IoT), and device-to-device communications (D2D). Network coding is a promising technology that can significantly improve the throughput and robustness of these emerging 5G multihop networks. However, network coding is generally very fragile to malicious attacks such as message content corruption and node compromise attacks. To take advantage of network coding in performance gain while refraining malicious network attacks is an interesting and challenging research issue. In this paper, we propose a new error-detection and error-correction (EDEC) scheme that can jointly detect and remove the malicious attacks based on the underlying error-control scheme for general multihop networks that can model the 5G multihop networks. The proposed scheme can increase the throughput for network with pollution attacks compared to existing error-detection based schemes. Then we propose a low-density parity check (LDPC) decoding based EDEC (LEDEC) scheme. Our theoretical analysis demonstrates that the LEDEC scheme can further increase the throughput for heavily polluted network environments. We also provide extensive performance evaluation and simulation results to validate the proposed schemes. This research ensures the expected performance gain for the application of network coding in the 5G network under malicious pollution attacks.

1. Introduction

Most of the newly emerging networks in the 5G communication network can be modeled using multihop networks, including vehicular ad hoc networks (VANETs), Internet of Things (IoT), and device-to-device communications (D2D). New computing frameworks in the 5G network such as fog computing [1] can also be categorized into multihop networks. The network throughput and robustness could be improved when network coding is utilized. Many researchers have proposed to adopt network coding in the upcoming 5G network, such as in [2–4]. The core notation of network coding is that it allows the participating nodes to encode incoming packets at intermediate network nodes in a way that when a sink receives the packets, it can recover the original message. Network coding provides a trade-off between maximum multicast flow rate and computational complexity.

Network coding was first introduced in the seminal paper by Ahlswede *et al*. [5]. Li *et al*. [6] formulated the multicast problem in network coding as the max-flow from the source to each receiving node. They proved that linear coding is sufficient to achieve the optimum. This work made network coding simpler and more practical. Koetter and Medard [7] have shown that linear codes are sufficient to achieve the multicast capacity by coding on a large enough field. Ho *et al*. [8] have shown that using of random linear network coding is a more practical way to design linear codes. Gkantsidis and Rodriguez [9] have applied the principles of random network coding to the context of peer-to-peer (P2P) content distribution and have shown that file downloading times can be reduced. Esmaeilzadeh *et al.* [10] proposed to use feedback-free random linear network coding in broadcasting layered video streams over heterogeneous single-hop wireless networks. Wu *et al.* [2] designed an efficient data dissemination protocol for VANETs using network coding. Lei *et al.* [3] applied network coding in Named Data Networking (NDN) in the 5G IoT network and improved the content delivery efficiency. Chi *et al.* [4] proposed to jointly utilize D2D communication and network coding to achieve high communication reliability in the ultra-dense 5G cell deployment.

However, in the context of network coding, all participating nodes must encode the incoming packets according to a fixed coding algorithm. If a packet from an intermediate relay node is corrupted or being tampered, the entire communication may be disrupted.

The main purpose of this paper is to develop schemes that can combat network pollution and malicious attacks from the network nodes based on error-control coding in multihop networks. We propose a new scheme that combines error-detection and error-correcting (EDEC) to combat network pollution attacks. In our scheme, the original message symbol is first encoded using an error-control code before encoded by network coding and transmitted. The application of error-control code gives intermediate nodes the capability to detect possible errors or pollution of the message. Unlike the existing schemes, when an intermediate node detects an error, the packet will continue to be forwarded. As long as the errors are correctable, the sink nodes will be able to recover the corrupted packets and decode the original packet symbols. Then we further extend the EDEC scheme and propose LDPC based EDEC (LEDEC) scheme. In the LEDEC scheme we treat the packets as LDPC codewords at the sink nodes and use the belief propagation algorithm (BPA) to decode the LDPC code. It can guarantee a certain network throughput even for a heavily polluted network environment, while the throughput becomes very low for error-detection based schemes. Moreover, we mainly focus on the throughput impact brought by different strategies (discard versus keep) towards corrupted packets.

The major contributions of this paper are the following:

(i) We propose an EDEC scheme by combining a modified error-control code and network coding. The proposed EDEC scheme can increase the throughput for network environment with pollution attacks compared to existing error-detection based schemes.

(ii) We propose the LEDEC scheme by augmenting the EDEC scheme with the LDPC decoding. The proposed LEDEC can further improve the throughput even for network environment with heavy pollution.

(iii) We conduct extensive simulations to evaluate the performance of the proposed schemes and demonstrate the advantage of the proposed schemes.

The rest of this paper is organized as follows: Section 2 presents the related work. Preliminaries for network coding, error-control coding and the proposed modified error-control code are discussed in Section 3. Adversary model is also presented in Section 3. The EDEC scheme and performance analysis are described in Section 4. Section 5 presents the LDPC decoding and analysis of the LEDEC scheme. We conclude in Section 6.

2. Related Work

Existing work on pollution elimination can largely be divided into error-detection based schemes and error-correction based schemes. For error-detection based schemes, the errors are normally detected at the intermediate forward nodes. For error-correction based schemes, the errors are generally corrected at the sink node.

While the error-correction based schemes seem to be more appealing, the computational complexity for encoding and decoding is relatively high [11]. These schemes are generally designed based on knowledge of the network topology, which makes these schemes less flexible to the current networks. Jaggi *et al.* [12] introduced the first polynomial-time rate-optimal network codes to correct the errors brought by malicious nodes. They also gave the theoretical network capacity with the existence of malicious nodes. However, the communication overhead of the scheme is significant. In [13], the authors proposed a network error-correcting code that combines the nonlinear coding at the source node and linear coding at the intermediate nodes. The code can achieve higher transmission rate than linear network error-correcting codes with exponential encoding and decoding complexity.

The limitations of error-correction based schemes make error-detection attractive in some network scenarios. Krohn *et al.* [14] proposed to use homomorphic hash functions to guarantee correctness of the network flow. The main idea is that each intermediate node will check the correctness of the packets through homomorphic hash functions. If a packet fails the correctness check at an intermediate node, it will be discarded. This approach can reduce the communication overhead and can be used in random network coding. However, the computational complexity is still very high. When the network scale is large, computing too many hash values will also create high delay. Boneh *et al.* [15] provided cryptographic protection against pollution attacks by authenticating linear subspaces in network coding, which incurs less computation delay than [14]. Shang *et al.* [16] proposed new homomorphic signature schemes for generation-based network coding.

In [17–20], homomorphic message authentication codes (MAC) were designed to detect polluted packets. Yu *et al.* [17] applied multiple MACs to each packet in secure XOR network coding to filter out polluted packets. Agrawal *et al.* [18] also designed a homomorphic MAC to check the integrity of network coded data with less computational overhead. Li *et al.* [19] proposed to use a symmetric key based homomorphic MAC algorithms to detect corrupted packets. In [21], the authors analyzed and improved two homomorphic authentication schemes homomorphic subspace signature (HSS) [22] and key predistribution-based tag encoding (KEPTE) [23] for network coding.

To further reduce computational and communication overhead, Kehdi *et al.* [24] developed a simple error-detection based *Null Key* scheme. Their idea is to partition the n-dimensional linear space over $GF(q^n)$ into two orthogonal subspaces of dimension k (symbol subspace) and $n-k$ (null key space). Comparing to the homomorphic hash function, the null key scheme is more efficient and has virtually no message delay. Newell *et al.* [25] improved the work of [24] by splitting the null key into one large constant portion and another small periodically updated portion, making it more suitable for network coding in wireless environments.

For all these error-detection based schemes, corrupted packets will be discarded. In packetized networks, a large message is divided into small packets. If a malicious node is able to continuously corrupt even a small fragment (packet) of the message, this fragment will be discarded. As a result, the whole message will be corrupted and the effective throughput will become lower and even close to zero in extreme situations. In our previous work [26], we proposed to keep the corrupted packets instead of dropping them; thus we could decode the corrupted packets in the intermediate forwarding nodes based on the subspace properties. In this paper we further generalize the idea and combine both error-detection and error-correction to guarantee a high throughput even for heavily polluted network environment.

It is worth noting here that the authors in [27] proposed using a cooperative nonparametric statistical framework (COPS) for misbehavior mitigation in network coding. Their framework does not need the knowledge of the data packet contents and provides another possible way to combat the network pollution attacks.

3. Preliminaries

In this section, we will first introduce basic concepts and notations of network coding. Then we will present the preliminary for error-control code and our proposed modified error-control code, which is the base of our proposed schemes. At last we will present our adversarial model adopted in this paper.

3.1. Network Coding. The main idea of network coding can be illustrated through the butterfly graph [6] in Figure 1. Assume the capacity of all the edges is C; the capacity of this network is $2C$ according to the max-flow min-cut theorem. Only by encoding the incoming bits x_1, x_2 at node 3, this network can achieve the maximum capacity.

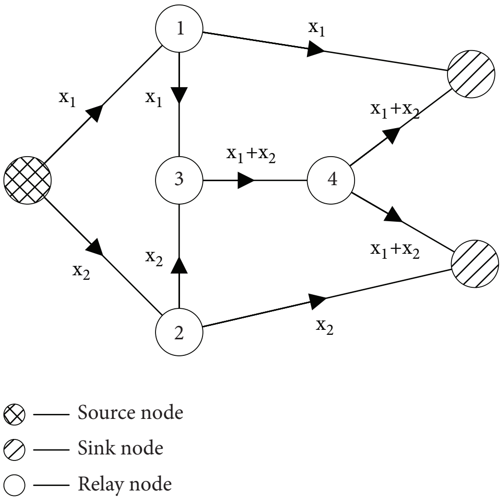


Figure 1: A simple example of network coding.

In this paper, we adopt the notations of [7]. A network is equivalent to a directed graph $G = (V, E)$, where V represents the set of vertices corresponding to the network nodes and E represents all the directed edges between vertices corresponding to the communication link. The start vertex v of an edge e is called the tail of e and written as $v = tail(e)$, while the end vertex u of an edge e is called the head of of e and written as $u = head(e)$.

For a source node u, there is a set of symbols $\mathscr{X}(u) = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ to be sent, where $\mathbf{x}_i \in GF(2^n)$, $1 \le i \le l$. For a link e between relay nodes r_1 and r_2, written as $e = (r_1, r_2)$, the symbol $\mathbf{y}_e$ transmitted on it is the function of all the $\mathbf{y}_{e'}$ such that $head(e') = r_1$. And y_e can be written as

$$\mathbf{y}_e = \sum_{e':head(e')=r_1} \beta_{e',e} \cdot \mathbf{y}_{e'} = \sum_{i=1}^{l} \beta_{e,i}\mathbf{x}_i = \boldsymbol{\beta}_e \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_l \end{bmatrix}, \qquad (1)$$

where $\beta_{e',e} \in GF(2)$ is the local network encoding coefficient, $\beta_{e,i} \in GF(2)$ is the global network encoding coefficient for symbol $\mathbf{y}_e$, and $\boldsymbol{\beta}_e = [\beta_{e,1}, \beta_{e,2}, \ldots, \beta_{e,l}]$ is the network encoding vector. For a sink node v, there is a set of incoming symbols $\mathbf{y}_1, \ldots, \mathbf{y}_m$ from e' where $tail(e') = v$ to be decoded.

For network coding to achieve the expected benefits, all the participating nodes in the network should be free of network pollution and malicious attacks. Suppose, under the linear network coding, the sink node receives m packet symbols $\mathbf{y}_1, \ldots, \mathbf{y}_m$. It decodes the original message symbols $\mathbf{x}_1, \ldots, \mathbf{x}_l$ by solving a set of linear equations:

$$\begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_m \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_l \end{bmatrix} = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1l} \\ \beta_{21} & \cdots & \beta_{2l} \\ \vdots & \ddots & \vdots \\ \beta_{m1} & \cdots & \beta_{ml} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_l \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix}. \qquad (2)$$

For clarity purpose, we number the network coding vectors $\boldsymbol{\beta}_e$ received in the sink node from 1 to m. If all the relay nodes encode correctly and $m \ge l$, the sink node can decode all the message symbols successfully.

3.2. Error-Control Coding

3.2.1. Error-Detection. Suppose the original message symbols are in the k-dimensional linear space over $GF(2^k)$. After we encode the symbols using the $k \times n$ generating matrix $\mathbf{G}$ of an (n, k) block code, the encoded codewords will form a linear subspace over $GF(2^n)$ of dimension k. So there will be another $n-k$ dimensional subspace over the n dimensional space, which is orthogonal to the codewords subspace. Denote a valid codeword by $\mathbf{c}$ and the bases for the $n-k$ dimensional subspace by $\mathbf{h}_1, \ldots, \mathbf{h}_{n-k}$; we have $<\mathbf{c}, \mathbf{h}_i>= 0$, $1 \leq i \leq n-k$, where $<\cdot, \cdot>$ represents the inner product.

Let $(n-k) \times n$ matrix $\mathbf{H} = [\mathbf{h}_1^T, \ldots, \mathbf{h}_{n-k}^T]^T$; $\mathbf{H}$ forms a parity check matrix of the codewords and satisfies

$$\mathbf{c} \cdot \mathbf{H}^T = \mathbf{0}. \tag{3}$$

Suppose $\mathbf{r} = \mathbf{c}+\mathbf{e}$ is a received codeword, where $\mathbf{e}$ is an n-tuple error generated by a malicious node. For the received word $\mathbf{r}$, we can get the *syndrome* of error pattern $\mathbf{e}$ as follows:

$$\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = (\mathbf{c} + \mathbf{e}) \cdot \mathbf{H}^T = \mathbf{c} \cdot \mathbf{H}^T + \mathbf{e} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T. \tag{4}$$

For a received codeword $\mathbf{r}$, there are two cases: (i) $\mathbf{e}$ equals to a legitimate codeword generated by $\mathbf{G}$. In this case, although $\mathbf{r}$ contains error, the error is undetectable using conventional error-control coding techniques; (ii) $\mathbf{e}$ contains a nonzero projection to the orthogonal parity check subspace, which satisfies $\mathbf{r} \cdot \mathbf{H}^T \neq 0$. In this case, we can detect that the received word contains error.

In network coding, suppose $\mathbf{c}_j$'s are a set of valid codewords, $\mathbf{c} = \sum_j \beta_j \mathbf{c}_j$ is a linear combination of these codewords, where β_j is the network encoding coefficient. It can be easily verified that for $1 \leq i \leq n-k$,

$$\mathbf{c} \cdot \mathbf{h}_i^T = \sum_j \beta_j \mathbf{c}_j \cdot \mathbf{h}_i^T = 0. \tag{5}$$

Equation (5) ensures that we can still check the correctness of packet symbols using row vectors of $\mathbf{H}$ after they are encoded by network coding. Similar to [24], we call the row vectors $\mathbf{h}_i$, $1 \leq i \leq n-k$ null keys.

3.2.2. Error-Correction. From (4), it is clear that $\mathbf{r}$ is a codeword if and only if $\mathbf{s} = 0$. The task of maximum likelihood decoding is to find the minimum weight error pattern $\mathbf{e}$ such that $\mathbf{r} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T$. In this case, the received $\mathbf{r}$ is corrected to $\mathbf{r} + \mathbf{e} = \mathbf{c}$.

In linear network coding, although the packet symbols are not the original ones sent from source nodes, we can still perform error-correction using (4). Suppose $\mathbf{r}_1, \ldots, \mathbf{r}_i$ are received codewords from i incoming edges; $\mathbf{e}$ is the error vector added to the network encoding $\mathbf{r} = \sum_j \beta_j \mathbf{r}_j$ and $\mathbf{r}' = \mathbf{r} + \mathbf{e}$. If the error is within the correction capability of the error-control code, the syndrome will still be $\mathbf{r}' \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T + \sum_j \beta_j \mathbf{r}_j \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T + \mathbf{0} = \mathbf{e} \cdot \mathbf{H}^T$. Thus we can correct the error using syndrome decoding.

The error-detection and error-correction capabilities are determined by the (n, k) code structure. Based on the pollution levels of the network, we can select the error-control codes accordingly base on the following proposition.

Proposition 1 (see [28]). *For an (n, k) block code with the minimum distance D, it can detect all the $D-1$ or less errors, or it can correct all the $\lfloor (D-1)/2 \rfloor$ or less errors.*

In our proposed schemes, the corrupted packets detected at the intermediate nodes will not be discarded. Both the intact and corrupted packets will be gathered by the sink nodes. The sink nodes can correct the corrupted packets and have a higher communication throughput than the error-detection based schemes.

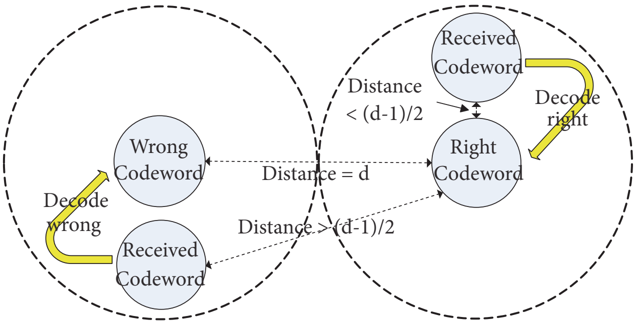


FIGURE 2: Limitations of error-control codes.

3.3. Modified Error-Control Code. The error-detection based schemes mainly focus on detecting the corrupt packets. As mentioned in Section 2, when a corrupt packet is detected, it will be discarded. So if an adversary continues to corrupt certain packets, these packets will be continuously dropped and the communication may never succeed. In this paper we try to utilize the corrupted packets to improve the throughput in these situations. To achieve this goal, we need to propose a modified error-control code.

3.3.1. Limitations of Conventional Error-Control Code. A linear error-correcting code encodes the original k bits message symbol $\mathbf{m}$ to an n bits codeword $\mathbf{c}$ using a $k \times n$ generating matrix $\mathbf{G}$. Suppose the minimum distance is D, according to the results in Section 3.2.2, the maximum number of errors we can correct is $\lfloor (D-1)/2 \rfloor$. If the number of errors is more than this amount, we may correct the corrupted codeword into a false one, as illustrated in Figure 2.

3.3.2. Proposed Error-Control Code That Can Detect Erroneous Decodings. The conventional error-control code may have undetected decoding errors. This is an inherent nature. No matter how low we set the code rate, these undetected errors may still exist. The decoding errors can only be detected using mechanisms other than a stand-alone error-correcting code.

Therefore, we propose to apply modified error-control code to both message symbols and network coding coefficients in (1). In this section, we will use the message symbol as an example. The original message symbol $\mathbf{m}$ is first mapped to a t bit value $\mathbf{h}$ using homomorphic hashes. Then $\mathbf{h}$ will be appended to $\mathbf{m}$ to form a new $k+t$ bits message symbol. By encoding this new message symbol we can get the final codeword. So the code becomes an $(n, k+t)$ code. By adding the extra bits, we can mitigate limitations of the conventional error-control code. Figure 3 illustrates the modified encoding scheme.

At the sink nodes, the decoded symbol is first split into two parts $\mathbf{m}'$ and $\mathbf{h}'$ after the decoding as shown in Figure 4. Then we calculate the mapping $\mathbf{h}''$ from $\mathbf{m}'$. If $\mathbf{h}''$

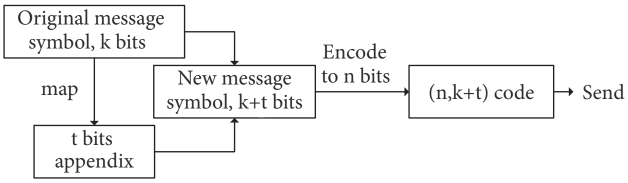


Figure 3: The encoding process of modified error-control code in EDEC scheme.

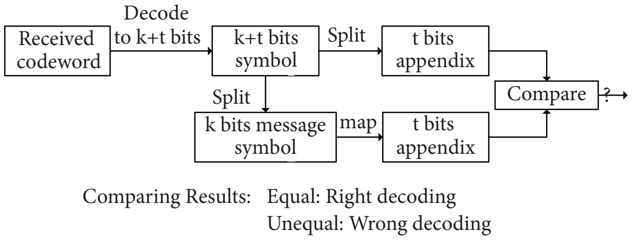


Figure 4: The decoding process of modified error-control code in EDEC scheme.

and $\mathbf{h}'$ are different, we can detect a decoding error. Our modification is equivalent to choose 2^k message symbols from 2^{k+t} symbols. Other message symbols in the 2^{k+t} symbol space are considered to be illegal. Because the decoding algorithm only guarantees that the decoded codeword is in the $k + t$ dimensional subspace, the decoded codeword belonging to the $2^{k+t} - 2^k$ symbol space implies that the decoding contains error.

Theorem 2. *Suppose a decoding error occurs, the wrong codeword will be any codeword in the 2^{k+t} symbol space. So the probability of detecting an erroneous decoding is*

$$p = \frac{2^{k+t} - 2^k}{2^{k+t}} = \frac{2^t - 1}{2^t} = 1 - \frac{1}{2^t}. \qquad (6)$$

Through properly choosing the parameters k, t, we can detect erroneous decodings with little additional overhead.

3.4. Adversary Model. In network coding, a small error injected into an intermediate relay node may diffuse to many packets at the sink node, which will incur errors in the message symbols after the solving of (2). This can cause a significant waste of network resources and sometimes can even ruin the whole network communication. This kind of attack is called *pollution attack*.

In this paper, we assume that the malicious node can inject bogus packets into the network. If the succeeding relay nodes do not detect the bogus packets and produce network coded packets using the bogus packets, we call these packets *corrupted packets*. The malicious nodes try to forge legitimate packets that can pass the check in (5). In error-detection based schemes that use null keys to check the validity of packet symbols, the source node will randomly distribute s null keys in each intermediate relay node for packet checking. The malicious node can increase its successful attack probability by producing bogus symbols orthogonal to the space spanned by the s null keys it receives.

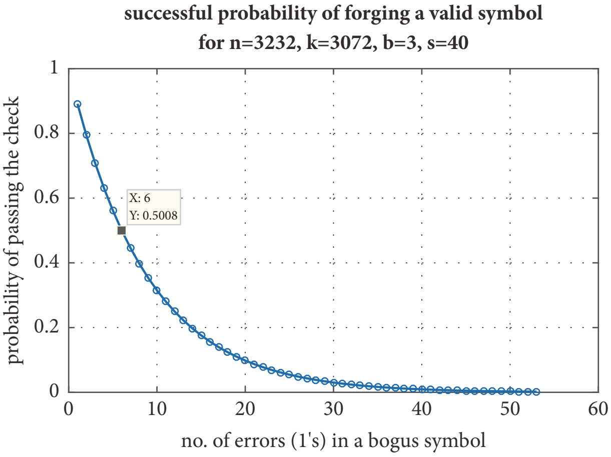


Figure 5: Probability for a forged symbol to pass the check.

When the size of the generating matrix is large, we will use sparse matrices $\mathbf{G}$ and $\mathbf{H}$ to reduce the encoding and decoding complexity. Suppose there are b 1's in each row and in each column on average in matrix $\mathbf{F} = \left[\begin{smallmatrix}\mathbf{G}\\ \mathbf{H}\end{smallmatrix}\right]$ for $b \ll n$, we have the following theorem.

Theorem 3. *The probability p_m that a malicious node successfully forges a valid symbol is*

$$p_m = \frac{\binom{k}{bw}}{\binom{n-s}{bw}}, \qquad (7)$$

where $\binom{n}{k}$ is the number of k-combinations from a set of n elements and $w \ll n$ is the number of 1's in a bogus symbol.

Proof (sketch). When the number of errors, which are 1's in the bogus symbol, is small compared to n, for each error bit with index i there will be b distinct vectors with 1's at the index i in $\mathbf{F}$. The bogus symbol will be linear dependent with these b vectors with high probability. Thus a bogus symbol with w errors ($w \ll n$) will be linear dependent with at most bw vectors, which may include rows from both matrices $\mathbf{G}$ and $\mathbf{H}$.

When a malicious node forges a symbol which is only linear dependent with the row vectors in $\mathbf{G}$, it will successfully forge a valid symbol. Since the malicious node does not know the row vectors of the matrices $\mathbf{G}$ and $\mathbf{H}$ except for the null key vectors it has, these row vectors can be viewed as random to the malicious node. The successfully probability is the number of ways of choosing bw vectors in all the row vectors of $\mathbf{G}$ divided by that of choosing bw vectors in all the row vectors of $\mathbf{F}$ excluding the s null key vectors the malicious node has already known. □

For large n, k, the malicious node has to make w small to achieve a high probability p_m. In a typical parameter setting with $n = 3232$, $k = 3072$, $b = 3$, $s = 40$, each intermediate node will receive 40 null keys randomly selected from 160 null keys. The successful probability of forging a valid symbol is shown in Figure 5. From the figure we can see that to achieve successful probability $p_m = 0.5$, the malicious node

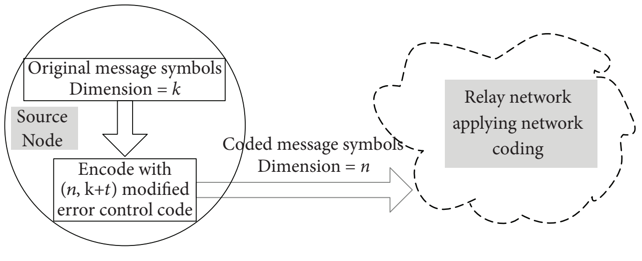


Figure 6: Apply error-control codes in linear network coding.

can only add 6 errors into a forged symbol, which are easily correctable by the (3232, 3072) code with minimum distance $D = 161$. *Thus for the malicious node, in order to produce bogus symbols that can pass the check, it will forge symbols with a small number of errors (1's) and orthogonal to the null key vectors it has.*

For the error-control code used to detect the bogus symbols, the errors within the bogus symbols are correctable. However, in the error-detection based schemes that do not perform error-correcting, the bogus symbols that pass the intermediate check will result in a failure of decoding the original message symbols at sink nodes.

4. Proposed EDEC Scheme and Evaluation

Similar to the null key scheme, our approach also utilizes the orthogonal space properties of the error-control code, but we use both the error-detection and error-correction techniques. The source node will encode the original message symbols using the modified error-control code prior to message transmission as shown in Figure 6. The properties of the error-control code keep unchanged during the linear network coding. When a corrupted packet is detected, instead of dropping it, we forward it to the sink nodes so that the packet can be used along with other packets to recover the original messages. However, the corrupted packet will not participate in network coding in the subsequent relay nodes once it is detected to be corrupted. The sink node can make use of the corrupted packets according to the decoding results of the modified error-control code in Section 3.3.

4.1. The Proposed EDEC Scheme. The proposed EDEC scheme is divided into two phases: initialization phase and transmission phase. The initialization phase is for null key and security parameter distribution while packet symbols are transmitted through network coding in the transmission phase.

4.1.1. Initialization Phase. In the initialization phase, the source node will first distribute s row vectors of the parity check matrix orthogonal to $\mathbf{G}$ in Algorithm 1 (null keys) to each of the relay nodes through a secret transmission protocol. Unlike normal linear network coding in which the network encoding vectors are attached to the start or the end of the packets, we propose to distribute the encoded encoding vectors to predetermined secret locations in the packets. The source node will send the location information to all the sink nodes during the initialization phase through the secure transmission protocol. This will prevent the malicious nodes from corrupting the encoding vector, which is essential for the data decoding. Moreover, the source node will also send the encoding matrix $\mathbf{G}_c$ for network encoding vectors and $\mathbf{G}$ for packet symbols to all the sink nodes. Once the initialization phase is done, the source nodes can multicast any number of packets to sink nodes. The overhead of the initialization phase is negligible.

4.1.2. Transmission Phase. In the transmission phase, the source node, relay nodes, and sink nodes will perform the proposed EDEC scheme according to Algorithms 1, 2, and 3.

In Algorithm 1, the source node will encode the network encoding vector $\boldsymbol{\beta}_i$ using the modified error-control code with a much longer appendix and a much lower code rate, compared to the encoding of packet symbols. This can improve the error resistance and detection probability for erroneous decodings to guarantee the correctness of the network encoding vectors used for data decoding. Since there is only one network encoding vector in each packet, the overhead brought by this higher security level is negligible.

Algorithm 2 presents the EDEC algorithm for relay nodes. Since the null key vectors are already distributed in the initialization phase, the relay nodes can check whether a packet is intact.

Algorithm 3 presents the EDEC algorithm for sink nodes. Since a sink node has already received the encoding matrix $\mathbf{G}_c$ and $\mathbf{G}$ in Algorithm 1 in the initialization phase, it can perform the error-control code decoding and detect the erroneous decodings. When collecting enough intact packets, it can derive the original data symbols through the decoding of network code.

4.2. Simulation Results of EDEC Scheme. In this section, we first present the simulation platform for EDEC scheme. Then we will compare the EDEC scheme and the error-detection based schemes, which are represented by the null key scheme in the simulation.

4.2.1. Simulation Platform. We simulate the EDEC scheme using ns-2. The scenario is set as a grid network with one source node, a number of relay nodes, and sink nodes. All the nodes are set as wireless nodes using wireless physical layer and 802.11 MAC protocol. The wireless channel is set to TwoRayGround. Once a node receives a packet, it will start the corresponding operations depending on its type (source, relay, malicious, and sink) and the packet content.

Figure 7 shows the topology of the simulated network. The source node is located at the lower left corner and 13 sink nodes lie at the upper right. The rest of the nodes are all intermediate nodes that can relay packets. In the simulation, we randomly choose a number of intermediate nodes as malicious nodes to perform pollution attacks. These nodes try to send out bogus packets that can pass the intermediate check as described in Section 3.4 to pollute the network. We can change the number of malicious nodes to evaluate performance of the algorithms under different network conditions. The percentage of malicious nodes is

```
for packet i do
    //Encode network encoding vector $\boldsymbol{\beta}_i$ in equation (1) using the modified error-control code (Figure 3)
    $\mathbf{h}_c \longleftarrow \mathrm{map}(\boldsymbol{\beta}_i)$
    $\mathbf{u}_c \longleftarrow (\boldsymbol{\beta}_i \mid \mathbf{h}_c)$
    Encoded network encoding vector $\longleftarrow \mathbf{u}_c \cdot \mathbf{G}_c$
    Distribute the encoded network encoding vector into predetermined locations of the packet
    for every symbol m of the packet do
        //Encode m using the modified error-control code (Figure 3)
        $\mathbf{h} \longleftarrow \mathrm{map}(\mathbf{m})$
        $\mathbf{u} \longleftarrow (\mathbf{m} \mid \mathbf{h})$
        Encoded symbol $\longleftarrow \mathbf{u} \cdot \mathbf{G}$
    end for
    Send out the encoded symbols
end for
```

Algorithm 1: EDEC algorithm for source nodes.

```
if every symbol in the received packet is intact then
    if $x$ (a predetermined number) packets received then
        Generate $x$ randomly, linearly combined packets using the received packets (network coding)
        Send out the network encoded packets
    end if
else
    Mark the packet as corrupted and send it out
end if
```

Algorithm 2: EDEC algorithm for relay nodes.

```
A packet is received
Decode every symbol in the packet using the decoding algorithm for the modified error-control code (Figure 4)
Reassemble the encoded network encoding vector from the predetermined secret locations and decode
  the network encoding vector
if the network encoding vector and all symbols are decoded correctly then
    if the packet is independent then
        Save the packet
        if $l$ (in equation (2)) independent packets are saved
        then
            Solve the network coding equations
        end if
    end if
end if
```

Algorithm 3: EDEC algorithm for sink nodes.

about 20% in Figure 7. The rest of the intermediate nodes act as relay nodes. After receiving a packet, they will first conduct pollution detection. In the error-detection based schemes, if the packet is corrupted, it will be dropped. In our proposed EDEC scheme, we will continue to forward the corrupted packet to the sink nodes. However, the corrupted packet will not participate in the succeeding network coding. The nodes behaviors will be detailed in the next section.

Because the packets are transmitted through broadcasting, although the MAC protocol is IEEE 802.11, we will still have packets collisions that will eventually influence the simulation results. In this paper, we only focus on evaluating the proposed EDEC and LEDEC schemes. According to the transmission range of a single node, adjacent nodes are assigned different time slots (see Figure 8) to avoid packets collisions. This can guarantee that the transmission range belonging to the same time slot would not overlap. We divide the entire time duration into 9 time slots and the duration of each time slot is 100 ms. The nodes are allowed to send packets only if they are in their own slots. If not, they will have to wait

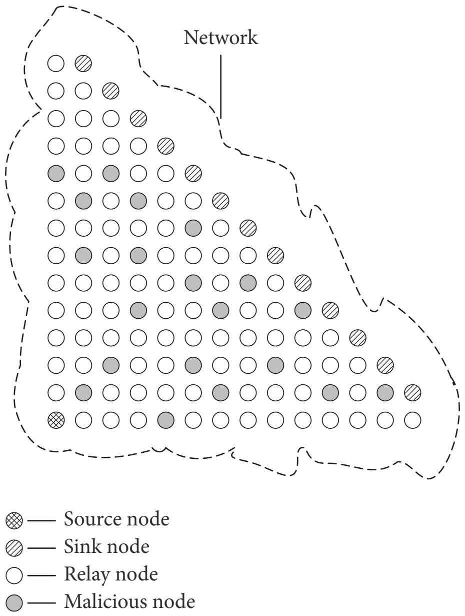


Figure 7: Simulation scenario.

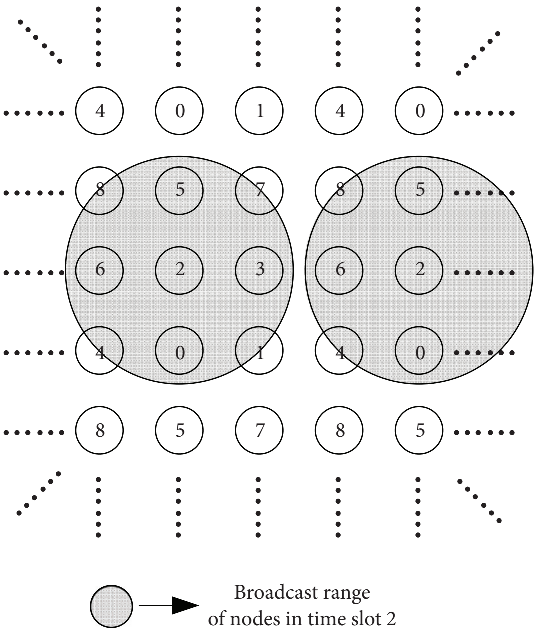


Figure 8: Illustration of the 9 time slots to avoid packets collisions.

until their next slots. In Figure 8 is an example in which the nodes belonging to time slot 2 can simultaneously transmit without packet collision.

4.2.2. Nodes Design. Four types of nodes are designed according to the algorithms described in Section 4.1 on the simulation platform.

(a) Source Node. In the simulation, the source node has $l = 32$ data packets to send. After initializing the network according to Section 4.1.1, the source node will encode the network encoding vector $\boldsymbol{\beta}_i$ for each packet using the modified error-control code presented in Figure 3 with $t = 32$. According to Theorem 2, the probability of detecting an erroneous decoding for the network encoding vector is $1 - 2^{-32}$, which means once the decoded network encoding vector passes the verification in Figure 4, we can view the network encoding vector as intact. Then the source node will distribute the encoded network encoding vector into predetermined locations in each packet. We fragment each packet embedding the encoded network encoding vector into symbols with the size 3056 bits and encode each packet symbol using the modified error-control code with $t = 16$. At last, the encoded packets are sent out into the network.

(b) Relay Nodes. Relay nodes will perform EDEC scheme according to Algorithm 2. Because the network is collision free and all the transmitted packets can be received, each packet only needs to be transmitted once. To utilize network coding efficiently while minimizing the transmission delay, each relay node will perform network encoding for every $x = 4$ valid packets it receiveds.

(c) Malicious Nodes. We assume that the malicious nodes will not perform network encoding. They only send out forged packets and conduct pollution attacks as described in Section 3.4.

(d) Sink Nodes. Sink nodes will decode both the modified error-control code and network code according to Algorithm 3.

4.2.3. Simulation Results. We conduct simulations under different percentages of malicious relay nodes. The throughput comparison between the EDEC scheme and the error-detection based schemes is shown in Figure 9. From the figure we can see that the EDEC scheme outperforms the error-detection based schemes in throughput: (i) When the percentage of malicious nodes is less than 10%, the two schemes have almost the same performance. (ii) With the increasing of malicious nodes, the performance of error-detection based schemes degrades significantly, while the throughput of the EDEC scheme remains almost unchanged. (iii) When the percentage of malicious nodes is larger than 40%, the throughput of the EDEC scheme begins to decrease because the cumulative number of errors in packet symbols becomes too large to correct by the modified error-control code. The uncorrectable packet symbols have the same degrading effect on the throughput as the packets discarded in the error-detection based schemes.

5. LDPC Decoding and LEDEC Scheme

In the EDEC scheme, only linearly independent packets participate in the network decoding at the sink nodes. Corrupted or linearly dependent packets will not be used, which is a waste of the resource. In this section, through utilizing these packets to recover more message symbols

```
while There are check nodes connected to only one unknown symbol node do
    for Each of these check nodes do
        The unknown symbol node ⟵ xor (All of the other symbol nodes connected to the check node)
    end for
end while
if All the unknown symbol nodes are recovered then
    Decode successfully
end if
```

Algorithm 4: BPA decoding algorithm for BEC.

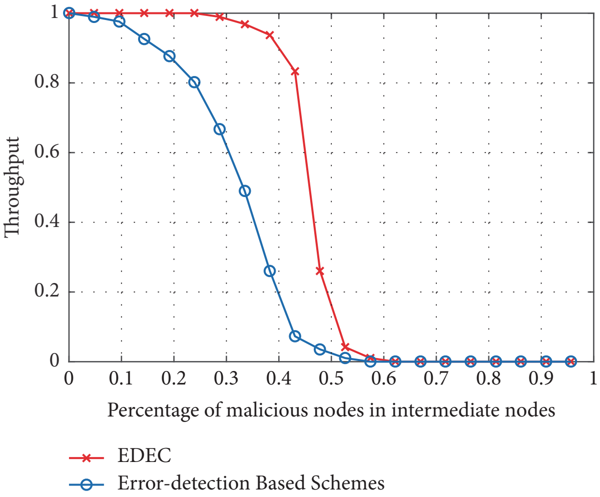


Figure 9: Throughput comparison between EDEC scheme and the error-detection schemes.

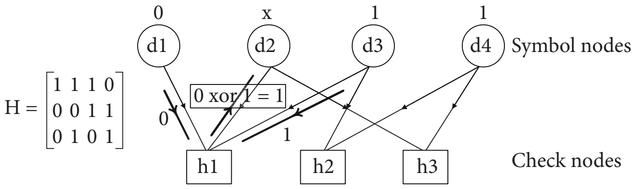


Figure 10: An illustrative example of parity check matrix and Tanner graph.

using low-density parity check (LDPC) decoding, we propose an LDPC decoding based EDEC (LEDEC) scheme.

5.1. LDPC Code. LDPC linear block code was first introduced by Gallager in 1962 [29]. One of the important characteristics of LDPC code is its sparse parity check matrix. By using iterative decoding, LDPC code can achieve error-correction performance close to Shannon bounds [30]. LDPC codes can be categorized as the regular LDPC code, of which the parity check matrix **H** has a fixed number of 1's per column and per row, and the irregular LDPC code, of which the parity check matrix may have different numbers of 1's in each column and each row. In this section, we will formulate the network coding to the irregular LDPC code.

5.2. Decoding of LDPC Code. The iterative decoding algorithm, known as belief propagation algorithm (BPA), is generally used to decode the LDPC code. Among all the channel models the BPA algorithm for the binary erasure channel (BEC) is the simplest, where only three numbers need to be considered: 0, 1, and x (erasure). The BPA can be described over the Tanner graph [31], which is a bipartite graph. In a Tanner graph, there are two types of nodes: the symbol nodes (corresponding to the received bits) and the check nodes (corresponding to the rows of the parity check matrix). An illustrative example of the parity check matrix and its Tanner graph is shown in Figure 10. In the parity check matrix, every row represents a parity check equation. The symbol nodes, which correspond to the bits equal to 1's in a row of the parity check matrix, are connected to the check node which corresponds to the same row. These nodes and edges in the Tanner graph express the parity check equation of that row. In Figure 10, node $h1$ represents the first row of the parity check matrix. The first, second, and third elements of the first row in parity check matrix are 1's, so symbol nodes $d1$, $d2$, and $d3$ are connected to $h1$ in the Tanner graph.

The decoding algorithm can be described through Algorithm 4 .

5.3. Relationship between Linear Network Code and LDPC Code. In linear network coding, packets are linearly combined at the intermediate nodes. The packets received at the sink nodes satisfy (2). In the network code decoding part of the EDEC algorithm, only independent valid packets are used. However, there is also helpful information in the linearly dependent packets or corrupted packets. If we can exploit and use these packets, the system performance can be further improved. Denote the received encoding vector as $\boldsymbol{\beta}_i = (\beta_{i,1}, \ldots, \beta_{i,l})$, where $1 \leq i \leq m$ and m is the number of received packets. Then the generation matrix of the block network code $\mathbf{G}_N$ can be defined as

$$\mathbf{G}_N = \left[\boldsymbol{\beta}_1^T \cdots \boldsymbol{\beta}_l^T, \boldsymbol{\beta}_{l+1}^T \cdots \boldsymbol{\beta}_m^T\right]^T. \quad (8)$$

As an example, suppose there is only one bit x_i in each original packet of the source node for $1 \leq i \leq l$. Define $\mathbf{x} = (x_1, \ldots, x_l)^T$. In this case, each received packet in the sink nodes also contains only one bit y_j for $1 \leq j \leq m$. Denote all the m received packets as a vector $\mathbf{y} = (y_1, \ldots, y_m)^T$. We have the following encoding equation:

$$\mathbf{y} = \mathbf{G}_N \mathbf{x}. \tag{9}$$

Theorem 4. *The linear network code can be viewed as a rateless LDPC code and can be decoded using the BPA algorithm.*

Proof (sketch). Since we can have an uncertain number of $m > l$ received encoded bits in the encoding equation (9), we can view the network code as a rateless code. The generating matrix $\mathbf{G}_N$ can be rewritten as

$$\mathbf{G}_N = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} I_l \\ P_2 P_1^{-1} \end{bmatrix} P_1, \tag{10}$$

where matrix P_1 can be made as an $l \times l$ full rank matrix through row exchange after l independent packets are received, P_2 is an $(m-l) \times l$ matrix, and I_l is an $l \times l$ identity matrix. The corresponding parity check matrix $\mathbf{H}_N$ can be written as

$$\mathbf{H}_N = \begin{bmatrix} \left(P_2 P_1^{-1}\right)^T \\ I_{m-l} \end{bmatrix}, \tag{11}$$

where I_{m-l} is an $(m-l) \times (m-l)$ identity matrix. We can verify the correctness of $\mathbf{H}_N$ by verifying the follow equation:

$$\mathbf{H}_N^T \mathbf{G}_N = \begin{bmatrix} \left(P_2 P_1^{-1}\right)^T \\ I_{m-l} \end{bmatrix}^T \begin{bmatrix} I_l \\ P_2 P_1^{-1} \end{bmatrix} P_1 = \mathbf{0}. \tag{12}$$

After deriving the corresponding parity check matrix $\mathbf{H}_N$, we can decode the linear network code using the BPA algorithm. The linear network code can be viewed as a rateless LDPC code and has the property of error-control codes. □

Although linear network codes can be viewed as rateless LDPC codes, the BPA algorithm cannot be used to decode a network code if the network code is derived after the decoding of a conventional error-control code, because we cannot detect the incorrect decodings which should be viewed as erasures. For the modified error-control codes in the EDEC scheme, we can determine the erroneous decodings and mark the corresponding bits as erasures. Therefore, we can decode the linear network code using the BPA algorithm. Figure 11 illustrates this main idea of the LEDEC scheme.

5.4. Theoretical Analysis. Through the information in linearly dependent packets, the LEDEC scheme can get additional benefits from BPA decoding of the LDPC code. Consider the case in which the percentage of malicious nodes is high, most of the packets are corrupted by the malicious nodes. The error-detection based schemes cannot work because all of the packets are discarded. The EDEC scheme does not work well either because with high erasure probability P_e there will not be enough correctable packets to solve the network coding equations (2).

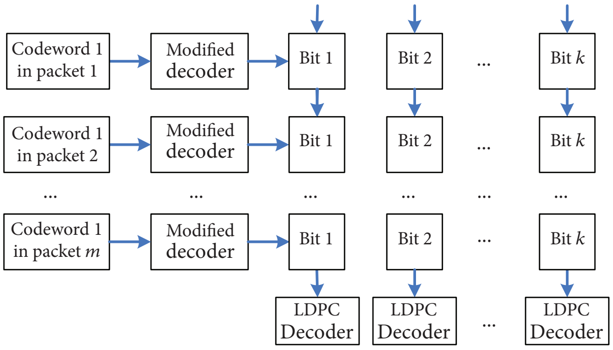


FIGURE 11: Main idea of the LEDEC scheme.

For the LEDEC scheme, let λ_d denote the probability that an edge from a check node is connected to a symbol node of degree d, and ρ_d denote the probability that an edge from a symbol node is connected to a check node of degree d in the Tanner graph of the corresponding LDPC code. The generating functions for an LDPC code is defined as: $\lambda(x) = \sum_d \lambda_d x^{d-1}$, $\rho(x) = \sum_d \rho_d x^{d-1}$. According to [32], the maximal fraction of erasures that a random LDPC code with given generating functions can correct is bounded by $P_{\max} = \min\{x/\lambda(1-\rho(1-x))\}$ $(0 < x < 1)$ with probability at least $1 - \mathcal{O}(n^{-3/4})$, where n is the length of the code. For the throughput of the LEDEC scheme, we have Theorem 5.

Theorem 5. *The throughput of the LEDEC scheme is*

$$F = \sum_{i=0}^{\lfloor N \cdot P_{\max} \rfloor} \binom{N}{i} P_e^i \left(1 - P_e\right)^{N-i}, \tag{13}$$

where $P_{\max} = \min\{x/\lambda(1-\rho(1-x))\}$ $(0 < x < 1)$, P_e is the erasure probability, N is the number of packets a sink node received, and $\lfloor \cdot \rfloor$ is the floor function.

Proof. Suppose a sink node receives N packets and the erasures in the packets are independent; the distribution of the number of erasures i in N received packet symbols is a binomial distribution with $\Pr(i) = \binom{N}{i} P_e^i (1-P_e)^{N-i}$, $0 \leq i \leq N$ as the probability mass function (PMF).

The proposed scheme can combat all erasures up to $N \cdot P_{\max}$ with probability at least $1 - \mathcal{O}(N^{-3/4})$, which is close to 1. Thus the throughput can be written as $F = \sum_{i=0}^{\lfloor N \cdot P_{\max} \rfloor} \Pr(i)$. □

5.5. Performance Analysis and Simulation. In this section, we provide simulation results of the LEDEC scheme on the simulation platform presented in Section 4.2. All the settings and parameters are the same as in Section 4.2.

5.5.1. Nodes Design. For the LEDEC scheme, the source node, relay nodes, and malicious nodes are the same as those in Section 4.2. The decoding process in the sink nodes is different, in which all packets received will be used. The BPA decoding will not start until the sink nodes collect all

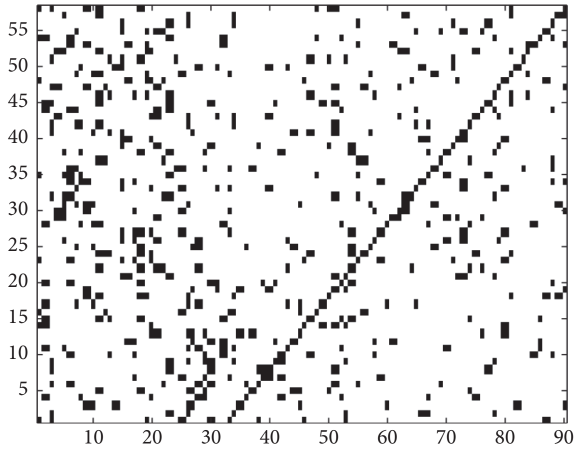


Figure 12: An example of the parity check matrix (transposed) in network coding.

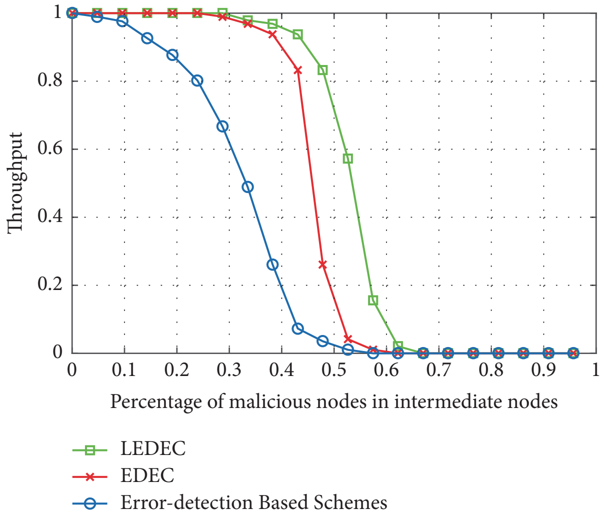


Figure 13: Performance evaluation of the LEDEC scheme.

the $l = 32$ independent packets. After receiving $l = 32$ independent packets, we can use the BPA algorithm to decode whenever a new packet arrives. However, there is a trade-off in determining when to start the BPA algorithm. When the algorithm is performed too frequently, the computational overhead will be high. On the other side, if we do not start the BPA decoding until we have collected a large number of packets, the communication delay will be high. To get a trade-off, the sink nodes will trigger the BPA decoding upon the receiving of every 10 new packets. This process will continue until all the message symbols have been successfully decoded.

5.5.2. Simulation Results. Same as in Section 4.2.3, the simulations in this section are carried out under different percentage of malicious relay nodes. One example of the parity check matrix (transposed) generated in the linear network coding is shown in Figure 12. In this example, the sink node receives 90 packets and decodes the linear network code using the BPA algorithm. In the figure, white squares represent 0 and black squares represent 1. The performance of the LEDEC scheme is shown in Figure 13.

Remark 6. When the percentage of the malicious nodes is less than 40%, the performance of the LEDEC scheme is slightly better than the EDEC scheme. This is because the sink nodes can successfully correct most of the errors in the corrupted packets and decode the original symbols using error-free packet symbols.

Remark 7. Because the sink nodes can recover extra information from the corrupted packets, when the percentage of malicious nodes is between 40% and 60%, the LEDEC scheme outperforms the EDEC and the error-detection based schemes.

6. Conclusion

In the paper, we focus on combating pollution attacks in multihop networks that utilize network coding, which can model most of the newly emerging networks in the 5G network. Our purpose is to maintain the network throughput even when the percentage of malicious nodes is large (≥30%). We first introduce an error-detection and error-correction (EDEC) scheme. By utilizing the information available in the corrupted packets, the network throughput can be increased with only a slight increase of the computational overhead compared to the error-detection based schemes. To further increase the throughput for the network environment with heavy pollution, we introduce LEDEC scheme that enables channel information be exploited and belief propagation algorithm (BPA) be used for the packet symbol recovery. This scheme can guarantee the throughput under the heavy pollution (percentage of malicious nodes is larger than 40%). We formulate the throughput of the LEDEC scheme through theoretical analysis and conduct comprehensive simulations to evaluate the performance. Our extensive simulation results show that the LEDEC scheme achieves better throughput than the EDEC scheme.

Data Availability

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

Disclosure

Jian Li's first part of the work was done when he was with the Department of ECE, Michigan State University, East Lansing, MI 48824-1226.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities KWJB17052536 and in part by the National Natural Science Foundation of China under Grant 61701019. This work is based on the work of the PhD dissertation of Jian Li [33].

References

[1] W. Zhang, Z. Zhang, and H. Chao, "Cooperative fog computing for dealing with big data in the internet of vehicles: architecture and hierarchical resource management," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 60–67, 2017.

[2] C. Wu, X. Chen, Y. Ji, S. Ohzahata, and T. Kato, "Efficient broadcasting in VANETs using dynamic backbone and network coding," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6057–6071, 2015.

[3] K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang, "An ndn iot content distribution model with network coding enhanced forwarding strategy for 5g," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 2725–2735, 2018.

[4] K. Chi, L. Huang, Y. Li, Y.-H. Zhu, X.-Z. Tian, and M. Xia, "Efficient and reliable multicast using device-to-device communication and network coding for a 5G network," *IEEE Network*, vol. 31, no. 4, pp. 78–84, 2017.

[5] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[6] S. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[7] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[8] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proceedings of the International Symposium on Information Theory ISIT' 04*, p. 44, July 2004.

[9] C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in *Proceedings of the IEEE INFOCOM '06*, pp. 1–13, April 2006.

[10] M. Esmaeilzadeh, P. Sadeghi, and N. Aboutorab, "Random linear network coding for wireless layered video broadcast: general design methods for adaptive feedback-free transmission," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 790–805, 2017.

[11] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proceedings of the IEEE Information Theory Workshop (ITW '02)*, pp. 119–122, 2002.

[12] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," in *Proceedings of the IEEE INFOCOM 2007: 26th IEEE International Conference on Computer Communications*, pp. 616–624, USA, May 2007.

[13] W. Guo, D. He, and N. Cai, "On capacity of network error correction coding with random errors," *IEEE Communications Letters*, vol. 22, no. 4, pp. 696–699, 2018.

[14] M. N. Krohn, M. J. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, pp. 226–239, May 2004.

[15] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: signature schemes for network coding," in *Public Key Cryptography-PKC 2009*, vol. 5443 of *Lecture Notes in Comput. Sci.*, pp. 68–87, Springer, Berlin, Germany, 2009.

[16] T. Shang, T. Peng, Q. Lei, and J. Liu, "Homomorphic Signature for Generation-based Network Coding," in *Proceedings of the 2016 IEEE International Conference on Smart Cloud, SmartCloud 2016*, pp. 269–273, USA, November 2016.

[17] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient scheme for securing XOR network coding against pollution attacks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 406–414, April 2009.

[18] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in *Applied Cryptography and Network Security: 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2–5, 2009. Proceedings*, vol. 5536 of *Lecture Notes in Computer Science*, pp. 292–305, Springer, Berlin, Germany, 2009.

[19] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE authentication for network coding," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, March 2010.

[20] A. Le and A. Markopoulou, "Cooperative defense against pollution attacks in network coding using SpaceMac," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 442–449, 2012.

[21] C. Cheng, J. Lee, T. Jiang, and T. Takagi, "Security analysis and improvements on two homomorphic authentication schemes for network coding," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 993–1002, 2016.

[22] P. Zhang, Y. Jiang, C. Lin, H. Yao, A. Wasef, and X. Shenz, "Padding for orthogonality: Efficient subspace authentication for network coding," in *Proceedings of the IEEE INFOCOM 2011*, pp. 1026–1034, China, April 2011.

[23] W. Xiaohu, X. Yinlong, Y. Chau, and X. Liping, "A tag encoding scheme against pollution attack to linear network coding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 33–42, 2014.

[24] E. Kehdi and B. Li, "Null keys: limiting malicious attacks via null space properties of network coding," in *Proceedings of the 28th Conference on Computer Communications, IEEE INFOCOM '09*, pp. 1224–1232, April 2009.

[25] A. Newell and C. Nita-Rotaru, "Split Null Keys: A null space based defense for pollution attacks in wireless network coding," in *Proceedings of the 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2012*, pp. 479–487, Republic of Korea, June 2012.

[26] W. Qiao, J. Li, and J. Ren, "An efficient error-detection and error-correction (edec) scheme for network coding," in *Proceedings of the IEEE Globecom '11*, pp. 1–5, December 2011.

[27] A. Antonopoulos and C. Verikoukis, "COPS: cooperative statistical misbehavior mitigation in network-coding-aided wireless networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1436–1446, 2017.

[28] S. Lin and D. J. Costello, *Error Control Coding*, Prentice Hall, 2nd edition, June 2004.

[29] R. G. Gallager, "Low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 8, pp. 21–28, 1962.

[30] C. E. Shannon, "A mathematical theory of communication," *Bell Labs Technical Journal*, vol. 27, pp. 379–423, 1948.

[31] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[32] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient erasure correcting codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[33] J. Li, *Capacity assurance in hostile networks [Ph.D. dissertation]*, Michigan State University, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 6809796, 8 pages
https://doi.org/10.1155/2018/6809796

Research Article

Anonymous Communication via Anonymous Identity-Based Encryption and Its Application in IoT

Liaoliang Jiang,[1] Tong Li,[1] Xuan Li,[2] Mohammed Atiquzzaman,[3] Haseeb Ahmad,[4] and Xianmin Wang[1]

[1] *School of Computer Science, Guangzhou University, Guangzhou 510000, China*
[2] *College of Mathematics and Informatics, Fujian Normal University, Fujian 350000, China*
[3] *School of Computer Science, The University of Oklahoma, USA*
[4] *Department of Computer Science, National Textile University, Faisalabad, Pakistan*

Correspondence should be addressed to Tong Li; 1120140107@mail.nankai.edu.cn

Received 30 June 2018; Accepted 24 September 2018; Published 21 November 2018

Guest Editor: Karl Andersson

Under the environment of the big data, the correlation between the data makes people have a greater demand for privacy. Moreover, the world has become more diversified and democratic than ever before. Freedom of speech is considered to be very important; thus, anonymity is also a very important security demand. The research of our paper proposes a scheme which can ensure both the privacy and the anonymity of a communication system, that is, the protection of message privacy while ensuring the users' anonymity. It is based on anonymous identity-based encryption (IBE), by which the users' *metadata* are protected. We implement our scheme in JAVA with Java pairing-based cryptography library (JPBC); the experiment shows that our scheme has significant advantage in efficiency compared with other anonymous communication system. Internet-of-Things (IoT) involves many devices, and privacy of devices is very significant. Anonymous communication system provides a secure environment without leaking metadata, which has many application scenarios in IoT.

1. Introduction

In the era of big data, data privacy has become significant as more personal and organizational information is involved. Moreover, the world has become more diversified and democratic than ever before. Following this trend, researchers have unveiled various ways through which adversaries can access private or otherwise sensitive information by network breach such as a called telephone number [1] or the IP address [2]. Therefore, the ultimate objective of privacy is to protect not only the contents of the messages but also the identities of communication parties, the actual time of a communication, and specific user participation during communication. For the above reasons, the research on privacy protection protocols such as anonymous communication system is imperative. In an anonymous communication system, the adversary must not know participant's identity at any time. Further, the adversary should not know the sending and receiving entities and whether a message is valuable. Overall, such pieces of information are called *metadata*. The *metadata* involves crucial private data in the anonymous communication system and it is also a critical parameter of the monitoring for the government and other stakeholders [3]. Anonymous communication system is very practical, and it can be applied to many realistic scenarios. Cloud computing has received more and more attention in recent years, and privacy-preserving and security under cloud environment have become critical issues [4–7], so high-level anonymity is required. In the neural network [8, 9] and secure deduplication [10], the anonymity is also critical.

Some private messaging systems are built on previous work [11–14]. However, such schemes do not provide efficient protocols without leaking *metadata*, which breaches the purpose of anonymous communication systems. There exist two categories of anonymous communication protocols. The first one is called Tor [15], which achieves the anonymity by encrypting the messages in layers under the public key cryptography. Several servers are employed in Tor, and the message is encrypted into a data packet by all servers' public key in turns (the construction of an onion). The encrypted data package is transmitted by the *chain* comprised of all servers. As a server receives a data package, it decrypts the data package through its private key and sends the decrypted data package to the next server; each server carries out the decryption operation until the data package is sent to the recipient. Tor is secure as each server knows only its predecessor and successor rather than other servers' position in the chain; therefore, the servers do not know the specific path of a message's transmission, which guarantees the anonymity of both the sender and recipient. That is how it offers a practical scheme of identity protection. However, some researchers showed that this scheme cannot defend against the traffic analysis attacks [16–18]. Tor's hidden service requires that every server must be honest; for instance, the security will be breached if the "first-server" and the "last-server" make a collusion. The second category is based on the DC-net [19, 20], for instance, verdict [21] and dissent [22]. Those schemes work on an N-member group, who can communicate anonymously, and only one group member is allowed to send a valid message in a given round. Each member shares a value secretly with the other N-1 members, which means that each member has N-1 secret values that are shared with the other N-1 members. Then, each member performs XORs operations with the N-1 shared secret values, but the legitimate sender performs the XORs with the N-1 shared secret values along with an additional message to generate ciphertexts that are later broadcast to the other members. Subsequently, each member receives all the ciphertexts and performs XORs operation on the N ciphertexts together to reveal the message. Because all the shared secret values had been XORed twice, they are cancelled out; thus, the message is extracted but without leaking the identity of the sender. In this way, the DC-net can prevent the traffic attack, as the DC-net is built on *anytrust* model that remains secure until at least one participating server remains honest. However, the recovery of messages must be computed by all the users, which is unrealistic scenario as none of the users could be off-line during underlying process. Hence, the existing methods are vulnerable to the internal dishonest member attacks which can easily break the security. Although, the systems may trace the dishonest member, they cannot exclude the influence of dishonest member during the communication. In summary, these two schemes above are of low-efficiency and the cost of communication is high.

In this paper, we propose an efficient anonymous communication scheme that offers higher security while disposing the aforementioned attack. The proposed scheme incorporates anonymous identity-based encryption (IBE) to achieve anonymous communication. In our scenario, more than one message can be sent simultaneously in each round. Meanwhile, each user uses its ID as the public key; we assume that the ID of each user is unique and it must be well known to other users. We set up a bulletin board from where every user could upload or download the ciphertexts directly at specific time during each round. The recovery of the message is based on the decryption by the recipient rather than the cooperations of all the users, so the system does not need the users to be on-line all the time during the communication, which is a remarkable advantage compared with the verdict and dissent. Since every user has to perform the same operation at the same time, the adversary cannot analyze who is the sender and who is the recipient in a given round. This characteristic ensures strong anonymity of the users.

The primary contributions of the paper are listed as follows:

(1) The user can send and receive the ciphertext during the same round, which means a user can be both a sender and a recipient at the same time. This can improve the efficiency of anonymous communication system.

(2) Our scheme allows the users to send or receive more than one valid "ciphertext" (because the message is extracted after decrypting the ciphertext) in each round. There is no limit to the number of communication ciphertexts. It is a huge advantage compared with other anonymous communication systems which are based on DC-net; this characteristic greatly improves the efficiency of anonymous communication system and reduces the cost of communication.

(3) In the proposed scheme, some dishonest users are tolerable in the communication, because a single user can successfully recover the message by himself instead of through the cooperation of all the other users.

Section 3 introduces the basic concepts of bilinear pairing, IBE and anonymous IBE. Section 4 outlines the system architecture and our security goals, and Section 5 describes the specific scheme and its two protocols. Section 6 presents a security analysis of the proposed system.

2. Related Work

Anonymous communication system has a high-level demand for security; we may consider a mechanism for entering into the anonymous communication system before starting communication. Users can do anonymous authentication before joining the system [23]; only a legal user can be a member of the system. Anonymous authentication can apply to many areas [24–27]. Besides, we can make an improvement of the storage pattern for users' messages. We may combine oblivious RAM [28, 29] with our anonymous communication system; oblivious RAM hides the access patterns of data, and it can prevent adversary from speculating sensitive information through users' access patterns. In the end, we may use other cryptographical techniques [30–33] to enhance the security of our anonymous communication system.

3. Preliminaries

This section outlines the anonymous IBE scheme [34], the difference between ordinary IBE and anonymous IBE. Although the identity-based encryption was firstly proposed by Shamir [35], a practical IBE scheme was constructed by Boneh and Franklin in 2001 [36], and after that, many IBE schemes were proposed [37–40]. The conception of anonymous identity-based encryption was firstly proposed in [41]. This section also introduces the basic knowledge of bilinear pairings, which are used to construct the scheme in the following sections.

3.1. Bilinear Map. Let G_1 be the additive cycle group generated by g and G_2 be multiplicative cyclic group. Prime p is the order of G_1 and G_2. The map $e : G_1 \times G_1 \longrightarrow G_2$ is called bilinear map, if it satisfies the following properties:

(1) Bilinearity: For any $P, Q \in G_1, a, b \in Z$, the following formula holds: $e(P^a, Q^b) = e(P, Q)^{ab}$.

(2) Nondegeneracy: The map e does not map all the pairs in the $G_1 \times G_1$ to the generator in G_2. If P is the generator of G_1, then $e(P, P)$ is the generator of G_2.

(3) Computability: For any P and Q, there is an algorithm that can compute $e(P, Q)$ efficiently.

3.2. IBE. In the IBE scheme, the participating parties include the users and the private key generator (PKG). The identity of the user is considered as the public key that makes IBE different from the traditional public key cryptography. The PKG, which is a trusted third party, generates the private key based on its master key and the user's identity. Subsequently, the private key is distributed to the corresponding user by the PKG. IBE is advantageous and is widely used for information security protection. Firstly, the key management is easy and efficient, because it does not require distributing public key or revoking the key. Secondly, IBE removes the certificate requirement for the public key of user who participates in the communication. For instance, when Alice wants to send a message to Bob, she uses Bob's public key ID_{Bob} that is known to each user. Alice encrypts the message with Bob's identity ID_{Bob} rather than querying Bob's public key from the PKI. This characteristic highlights the purpose of anonymous communication. Suppose that Alice queries Bob's public key from the PKI; the adversary can easily get information about sender/recipient through the operation of querying, so the security goals are breached. After Bob gets the encrypted message, he decrypts the message with his private key that he gets from the PKG. IBE consists of the following four functions [42]:

Setup: inputting security parameter k, returning the public parameter *params* and the master key *msk* of the system. The limited plaintext space M, the limited ciphertext space C, and the *params* are public, and the master key *msk* is secretly kept by the PKG.

Extract: inputting the master key *msk* and a user's identity *id*, generating the corresponding private key sk_{id} for *id*.

Encryption: inputting the message $m \in M$ and *id*, returning the ciphertext $c \in C$.

Decryption: inputting the ciphertext c and the corresponding private key sk_{id} for *id*, returning the plaintext m.

3.3. Anonymous IBE. In traditional IBE, since the recipient's identity is used as the public key, this property may leak the recipient's identity through the adversary's analysis of the ciphertexts. In other words, if the user's identity is leaked, the anonymous communication system is no longer secure. An anonymous IBE scheme must obey two properties:

(1) The adversary cannot get any information about the communication parties.

(2) The user's identity cannot be unveiled by the ciphertexts.

In this paper, we use the anonymous IBE scheme which is based on bilinear map [34]. Let G_1 and G_2 be the group of order p; the map $e : G_1 \times G_1 \longrightarrow G_2$ is the bilinear map, and g is the generator of group G_1. $\sigma \in Z_{p^*}$, $g_2 \in G_1$ are randomly selected, and let $g_1 = g^{\sigma}$. The scheme includes four functions as follows:

Initialization: choose the public parameters g, g_1, g_2 and the master key of PKG denoted as σ.

Private key generation: the PKG randomly selects $r \in Z_{p^*}$ and computes the private key for the corresponding recipient; here ID is the recipient's ID and $ID \in Z_{p^*}$.

$$d = (d_1, d_2, d_3) = \left(g_2^{\sigma} g_1^{ID \cdot r}, g^r, g^{ID \cdot r}\right) \tag{1}$$

Encryption: suppose a message $m \in G_2$ needs to be encrypted. The sender randomly selects $t, s \in Z_{p^*}$ and computes the ciphertext.

$$c = (c_1, c_2, c_3, c_4) = \left(e\left(g_1, g_2\right)^t \cdot m, g_1^{ID(t+s)}, g_1^s, g^t\right) \tag{2}$$

Decryption: the recipient uses his own private key $d = (d_1, d_2, d_3)$ to decrypt the ciphertext to obtain the plaintext as follows.

$$m = c_1 \cdot \frac{e\left(d_2, c_2\right)}{e\left(d_1, c_4\right) e\left(d_3, c_3\right)} \tag{3}$$

Suppose that an adversary wants to extract the recipient's identity ID from the ciphertext, it is obvious from the structure of the ciphertext that the ID can only be extracted through $c_2 = g_1^{ID(t+s)}$ by using $c = (c_1, c_2, c_3, c_4)$. Although g_1 is a public parameter, but the t, s are randomly selected from the Z_{p^*} and it is impossible for the adversary to obtain these parameters. Meanwhile, $ID(t + s)$ is the exponent of c_2; the computation is complicated.

4. Architecture of the System

We describe the details of system entities and system architecture in this section; furthermore, we present the security goals of anonymous communication.

4.1. Entities. (i) **The users.** The users are an imperative part of the system whose privacy must be assured, while the users can communicate with any user in the system. The system can satisfy two kinds of users. The first kind is those who just want to disclose some message anonymously; however, the users of this kind do not want to disclose the identity of the sender even to the recipient, for instance, a journalist wants to disclose a scandal about a politician who has participated in the presidential campaign. The second kind of users want to communicate with someone secretly; the users of this kind do not want anyone to know who is communicating with them, or whether he is involved in this communication. For instance, two executives from a tendering company want to negotiate about the final bidding price though geographical differences.

(ii) **Bulletin board.** The bulletin is provided to the users for uploading and downloading ciphertexts. More precisely, the sender uploads the encrypted message to the bulletin board, and the recipient downloads the message from the bulletin board. But as we mentioned above, the bulletin board is an intermediate source for communication, and there is no need for an interaction between users. Because there is no interaction between users, the adversary cannot know the identities of the communicating parties.

(iii) **Private key generator (PKG).** In the system, the role of PKG is to generate private keys of users against their IDs and we assume that PKG is honest.

4.2. Architecture. This section describes the architecture of the system. Each user has a unique identity. The system consist of two components.

(1) The Private Key Protocol. This is a protocol for distributing the private keys among the users according to their identity; this part works between the users and PKG. In order to carry out the decryption operation later, the user should obtain its private key before initiating a communication. The PKG generates the private key through users' identity and its master key for the corresponding users. For instance, Alice's identity is ID_{Alice} and Bob's identity is ID_{Bob}; the PKG generates private key sk_{Alice} for Alice and sk_{Bob} for Bob. Only by obtaining the private key, the later protocol can proceed. All the users should get their private key in specific time t_1.

(2) Communicating Protocol. It is the most important part of the system and it contains four operations. The user uses the recipient's identity as the public key to encrypt the message. For instance, Alice wants to send a message to Bob; Alice uses Bob's identity ID_{Bob} to encrypt the message. The encryption operation should be done in specific time t_2. For consideration of senders' anonymity, every user is required to send at least one message whether the user wants to have a communication or not. After the encryption operation, all the ciphertexts should be uploaded to the bulletin board. The uploading operation should be done in specific time t_3. In specific time t_4, each user downloads all the ciphertexts on the bulletin board. Since the users did not interact before and also the users do not know whether there is any message belonging to them in this round, all the messages in the bulletin board must be downloaded by each user in order not to miss the message.

All the users decrypt all the ciphertexts through its private key in specific time t_5. After time t_5, one round of communication during all the users is finished, and then the next round of communication can start. The subsequent communication rounds are scheduled as shown in Figure 1. In this paper, we have distinct advantage over the previous works. The advantage is that there is no limit to the number of messages sent by the user in each round, which means that one user can communicate with one or more users at a given round.

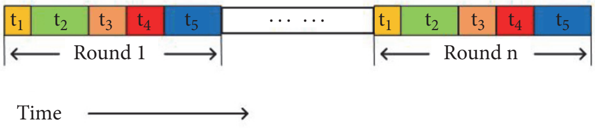


Figure 1: Scheduling of n rounds of communication.

4.3. Security Goal. The system ensures anonymous communication in three aspects.

(1) The message's security: The contents of users' message need to be protected, it is the basic requirement of a secure system.

(2) The sender's anonymity: Unauthorized users could not determine the identity of sender, so the adversary cannot judge which user sent a message in a given round. In other words, the identity of sender should not be leaked.

(3) The recipient's anonymity: The recipient's anonymity is to ensure that others cannot judge whether the message is received by a definite recipient. Furthermore, the system is required to guarantee that the adversary cannot extract the recipient's ID which is used during the process of encryption. Similar to sender's anonymity, the identity of recipient should not be leaked.

The sender's anonymity and the recipient's anonymity are the key security goals which are different from the traditional communication system, as the system should not reveal any *metadata* about users.

5. Anonymous Communication via Anonymous IBE

The scheme consists of two major components: private key generation and anonymous communication. In the presented solution, we construct a general scheme based on anonymous IBE which is provided in Section 3.3. G_1 and G_2 are the groups of order p, and g is the generator of group G_1. The map e is bilinear map which satisfies $G_1 \times G_1 \longrightarrow G_2$. $\sigma \in Z_{p^*}$ is the master key of PKG, $g_2 \in G_1$ are randomly selected, and let $g_1 = g^{\sigma}$. This section describes the construction of these two protocols.

5.1. Private Key Generation Protocol. The private keys are generated through the users' ID by PKG, then PKG distributes those private keys to corresponding users.

The details of private key generation protocol are described in Algorithm 1, where PKG randomly chooses an integer $r \in \mathbb{Z}_p^*$ and uses r to compute the private key k. After the computation, the PKG distributes the private key to the corresponding users.

5.2. Anonymous Communication Protocol. In our scheme, the anonymous communication protocol contains four steps as presented in Figure 2. The steps include encryption,

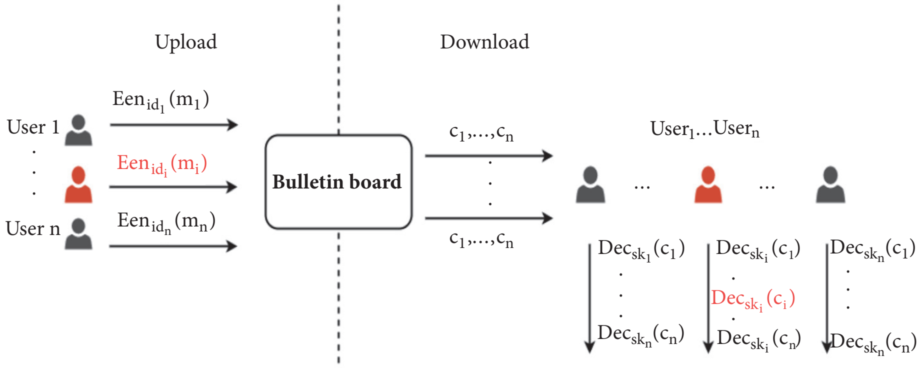

Upload
Download
User 1
User n
$\mathrm{Een}_{id_1}(m_1)$
$\mathrm{Een}_{id_i}(m_i)$
$\mathrm{Een}_{id_n}(m_n)$
Bulletin board
$c_1,\ldots,c_n$
$c_1,\ldots,c_n$
$\mathrm{User}_1\ldots\mathrm{User}_n$
...
...
$\mathrm{Dec}_{sk_1}(c_1)$
$\mathrm{Dec}_{sk_n}(c_n)$
$\mathrm{Dec}_{sk_i}(c_1)$
$\mathrm{Dec}_{sk_i}(c_i)$
$\mathrm{Dec}_{sk_i}(c_n)$
$\mathrm{Dec}_{sk_n}(c_1)$
$\mathrm{Dec}_{sk_n}(c_n)$

Figure 2: The four steps of anonymous communication protocol: ① Encryption, ② Upload, ③ Download, ④ Decryption.

Input: A user's identity *id*, the public parameters g, g_1, g_2, and the master key σ.
Output: The user's private key $k = (k_1, k_2, k_3)$.
The PKG performs the followings:
randomly chooses $r \in \mathbb{Z}_p^*$;
computes the private key $k = (k_1, k_2, k_3) = (g_2^{\sigma} g_1^{id\cdot r}, g^r, g^{id\cdot r})$;
sends the private key to the corresponding users at specific time t_1;

Algorithm 1: Private key protocol.

(1) **Encryption**. If the sender just wants to disclose the message anonymously rather than anyone knows where is message from even the recipient, he computes the ciphertext $c = (c_1, c_2, c_3, c_4) = (e(g_1, g_2)^t \cdot m, g_1^{id(t+s)}, g_1^s, g^t)$; If the sender wants the recipient to know about its identity, the signature $Sign_{Sender_{id}}$ should be attached with the message m, compute the ciphertext $c = (c_1, c_2, c_3, c_4) = (e(g_1, g_2)^t \cdot (m||Sign_{Sender_{id}}), g_1^{id(t+s)}, g_1^s, g^t)$. The **Encryption** operation should done in specific time t_2.
(2) **Upload**. All the users uploads their ciphertexts c to the bulletin board at the specific time t_3. Each user's ciphertexts c are stored in the bulletin board after this operation.
(3) **Download**. Each involved users are required to download all the ciphertexts which stored in the bulletin board, each user does the **Download** operation at the specific time t_4.
(4) **Decryption**. The user decrypts the ciphertexts c which obtained from the bulletin board in **Download** operation, one by one through the private key k in specific time t_5. The plaintext is computed as $m = c_1 \cdot e(d_2, c_2)/e(d_1, c_4)e(d_3, c_3)$. If there is a signature $Sign_{Sender_{id}}$ attached with the message m, then the user can get $m||Sign_{Sender_{id}}$.

Algorithm 2: Anonymous communication protocol.

uploading, downloading, and decryption. The protocol is performed collaboratively by the users and the bulletin board.

In the anonymous communication protocol, each user u needs to encrypt his message to generate the ciphertext $c = (c_1, c_2, c_3, c_4)$. He first randomly selects $t, s \in Z_{p^*}$. Here, m is the message which needs to be encrypted, *id* is the recipient's ID, $Sign_{Sender_{id}}$ is the signature of sender's identity. As we mentioned in Section 4.1, our system can satisfy two kinds of users. If the sender wants the recipient to know where the message is from, the signature can be attached together with the message m. Otherwise, the signature is not necessary to be transmitted. Subsequently, the encrypted message c can be uploaded to the bulletin board. At this step, each user is required to upload at least one encrypted message whether he wants to initiate a communication or not. However, if a user wants to communicate with more than one user, it is permitted to upload more than one ciphertext at the same time. When a user wants to get a message, he needs to download all the ciphertexts from the bulletin board without any interaction with the uploaders. Then, the user just needs to use his private key to decrypt the ciphertexts one by one. If the decryption is successfully performed, it means that the message belongs to the user. It should be noted that each user is required to download all the messages on the bulletin board. Since there are no interactions between the users and the bulletin board before the communication, each user has no idea whether the message belongs to him in the process of a communication. In this way, all the users have to participate in the uploading and downloading operations, while the adversary does not know which parties are really involved in a given round. The details of anonymous communication protocol are shown in Algorithm 2.

6. Security Analysis

As we mentioned in Section 4.3, there are three aspects of security goals needed to be achieved: the message's security, the sender's anonymity, and the recipient's anonymity, and we will analyze the system security as follows.

Every message is encrypted before uploading and the encryption scheme we used can ensure the message's security. The security of encryption scheme we used in our construction had been proved in [34], this encryption scheme can defend against an arbitrary CPA adversary while maintaining anonymity.

In traditional public key cryptography, there is usually a public key infrastructure (PKI), and the sender needs to query the recipient's public key before initiating a communication. In this process, the user who does the operation of querying is likely to be the sender who wants to initiate a communication, and the public key to be queried likely belongs to the recipient. In our scheme, the sender no more needs to query the recipient's public key (because the public key is recipient's identity which is known to each user). On the other hand, although the recipient's identity is used as the public key, the anonymous IBE ensures that the adversary cannot extract the recipient's identity from the ciphertext. Since all the users perform the operation of uploading at time t_3 and download the same amount of ciphertexts at time t_4, the adversary cannot know which user has the intention to participate in a communication through the operations of uploading and downloading. Obviously, our scheme can guarantee the anonymity of both the sender and recipient.

7. Evaluation

In this section, we evaluate the performance of our scheme, which has been implemented in JAVA with Java pairing-based cryptography library (JPBC). All experiments were conducted on a PC with a CPU 2.13 GHz, 6 GB of RAM. In our implementation, a message's length was set as 128 bytes, and the time consumption of uploading and downloading was ignored.

7.1. Computational Consumption. We implemented our scheme in one message and executed 1000 rounds. The computational consumption includes three operations: private key generation, encryption, and decryption. Under the fiber configuration of 100 Mpbs, the cost of uploading and downloading is negligible. We calculate the time cost as in Figure 3. It takes approximately 3.6×10^4 ms to perform the private key generation operation for 1000 rounds, 1.12×10^5 ms to perform the encryption operation for 1000 rounds, and 1.62×10^2 ms to perform the decryption operation for 1000 rounds.

7.2. Communication Consumption. Our scheme has no limit for the number of messages in a round, it is a significant advantage compared with other anonymous communication systems which can send only one message in a round. There are common scenarios; for example, a user wants to communicate with more than one person, or more than one user wants to send message. In the anonymous communication system which limits the number of messages, users have to wait for several rounds. But, in our scheme, all users can send an arbitrary number of messages in a round. This property enhances the efficiency of communication and reduces the cost of communication. Figure 4 shows the communication consumption of our scheme and the anonymous communication system which limits the number of messages.

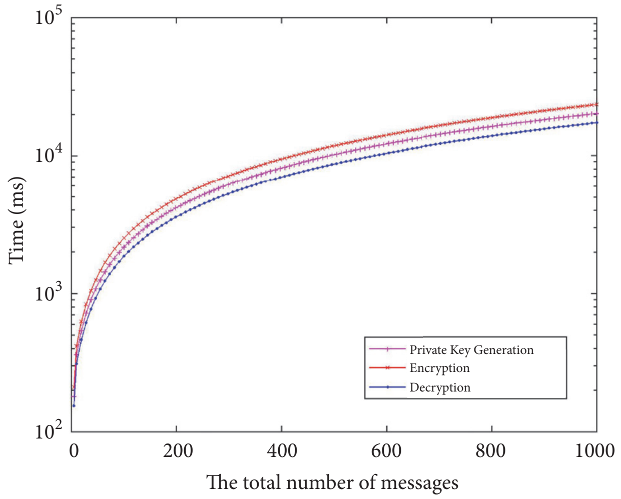


Figure 3: Computational consumption.

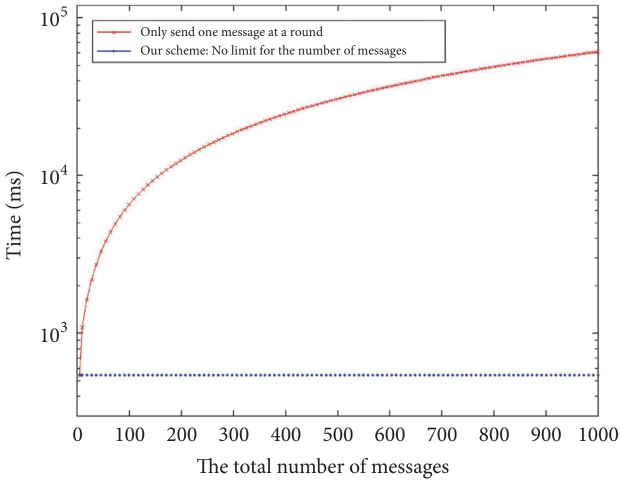


Figure 4: Communication consumption.

8. Conclusions

In this paper, we address a communication system which aims to protect the users' *metadata*. To solve this problem, we propose an anonymous communication system based on anonymous IBE. Our scheme has significant advantage in efficiency compared with previous work and can also offer strong anonymity. In the future, we will consider the user authentication and the application scenario in the smart environment.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts interests.

Authors' Contributions

The first author conducted the experiments and wrote the first draft of the paper. The other coauthors helped in revising the paper and polished it. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61472091), Natural Science Foundation of Guangdong Province for Distinguished Young Scholars (2014A030306020), Guangzhou Scholars Project for Universities of Guangzhou (No. 1201561613), and Science and Technology Planning Project of Guangdong Province, China (2015B010129015).

References

[1] J. Mayer, P. Mutchler, and J. C. Mitchell, "Evaluating the privacy properties of telephone metadata," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 113, no. 20, pp. 5536–5541, 2016.

[2] Y.-A. de Montjoye, *Computational PRIvacy: Towards PRIvacy-Conscientious Uses of Metadata*, ProQuest LLC, Ann Arbor, MI, 2015.

[3] A. Rusbridger, "The snowden leaks and the public," *New York Review of Books*, vol. 60, no. 18, 2013.

[4] P. Li, J. Li, Z. Huang et al., "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[5] C. Gao, Q. Cheng, X. Li, and S. Xia, "Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network," *Cluster Computing*, pp. 1–9, 2018.

[6] 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.

[7] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

[8] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51–62, 2018.

[9] C. Yuan, LiXinting, J. Q. M. Wu, j. Li, and X. Sun, "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Computers, Materials & Continua*, vol. 53, no. 3, pp. 357–371, 2017.

[10] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1615–1625, 2014.

[11] N. Borisov, G. Danezis, and I. Goldberg, "DP5: A Private Presence Service," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 4–24, 2015.

[12] D. Chaum, D. Das, F. Javani et al., "cMix: Mixing with Minimal Real-Time Asymmetric Cryptographic Operations," in *Applied Cryptography and Network Security*, vol. 10355 of *Lecture Notes in Computer Science*, pp. 557–578, Springer International Publishing, Cham, 2017.

[13] A. Kwon, D. Lazar, S. Devadas, and B. Ford, "Riffle: An efficient communication system with strong anonymity," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 2, pp. 115–134, 2016.

[14] J. Van Den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, "Vuvuzela: Scalable private messaging resistant to traffic analysis," in *Proceedings of the 25th ACM Symposium on Operating Systems Principles, SOSP 2015*, pp. 137–152, USA, October 2015.

[15] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," Defense Technical Information Center, 2004.

[16] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against Tor," in *Proceedings of the 6th ACM Workshop on Privacy in the Electronic Society, WPES'07, Held in Association with the 14th ACM Computer and Communications Security Conference*, pp. 11–20, USA, October 2007.

[17] S. Chakravarty, A. Stavrou, and A. D. Keromytis, "Identifying proxy nodes in a tor anonymization circuit," in *Proceedings of the 4th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2008*, pp. 633–639, Indonesia, December 2008.

[18] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" *ACM Transactions on Information and System Security*, vol. 13, no. 2, 2010.

[19] D. Chaum, "The dining cryptographers problem: unconditional sender and recipient untraceability," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 1, no. 1, pp. 65–75, 1988.

[20] P. Golle and A. Juels, "Dining cryptographers revisited," in *Advances in cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Comput. Sci.*, pp. 456–473, Springer, Berlin, 2004.

[21] H. Corrigan-Gibbs, D. I. Wolinsky, and B. Ford, "Proactively accountable anonymous messaging in verdict," in *Proceedings of the USENIX Security Symposium*, pp. 147–162, 2013.

[22] E. Syta, A. Johnson, H. Corrigan-Gibbs, S. Weng, D. Wolinsky, and B. Ford, "Security Analysis of Accountable Anonymous Group Communication in Dissent," Defense Technical Information Center, 2013.

[23] 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.

[24] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, and Z.-H. Zhan, "Secure data uploading scheme for a smart home system," *Information Sciences*, vol. 453, pp. 186–197, 2018.

[25] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Gao, “Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures,” *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[26] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, “Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks,” *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.

[27] Y. Zhang, J. Li, D. Zheng, P. Li, and Y. Tian, “Privacy-preserving communication and power injection over vehicle networks and 5 g smart grid slice,” *Journal of Network and Computer Applications*, vol. 122, pp. 50–60, 2018.

[28] B. Li, Y. Huang, Z. Liu, J. Li, Z. Tian, and S.-M. Yiu, “HybridORAM: Practical oblivious cloud storage with constant bandwidth,” *Information Sciences*, 2018.

[29] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, “DivORAM: Towards a practical oblivious RAM with variable block size,” *Information Sciences*, vol. 447, pp. 1–11, 2018.

[30] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, “A Covert Channel Over VoLTE via Adjusting Silence Periods,” *IEEE Access*, vol. 6, pp. 9292–9302, 2018.

[31] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, “An ID-based linearly homomorphic signature scheme and its application in blockchain,” *IEEE Access*, 2018.

[32] C. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, “Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack,” *Information Sciences*, vol. 444, pp. 72–88, 2018.

[33] Y. Zhang, D. Zheng, and R. H. Deng, “Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2130–2145, 2018.

[34] B. Wang and X. Hong, “An anonymous signature scheme in the standard model,” *Journal of Information Science and Engineering*, vol. 30, no. 6, pp. 2003–2017, 2014.

[35] A. Shamir, “Identity-based cryptosystems and signature schemes,” in *Advances in Cryptology: Proceedings of (CRYPTO ’84)*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Germany, 1985.

[36] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[37] B. Waters, “Efficient identity-based encryption without random oracles,” *Advances in Cryptology – EUROCRYPT 2005*, vol. 3494, pp. 114–127, 2005.

[38] D. Boneh and X. Boyen, “Secure identity based encryption without random oracles,” in *Advances in cryptology—CRYPTO 2004*, vol. 3152 of *Lecture Notes in Comput. Sci.*, pp. 443–459, Springer, Berlin, 2004.

[39] D. Boneh and X. Boyen, “Efficient selective identity-based encryption without random oracles,” *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 24, no. 4, pp. 659–693, 2011.

[40] J. Li, X. Chen, C. Jia, and W. Lou, “Identity-based encryption with outsourced revocation in cloud computing,” *IEEE Transactions on Computers*, 2013.

[41] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in *Advances in Cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 506–522, Springer, Berlin, Germany, 2004.

[42] D. Boneh and X. Boyen, “Secure identity based encryption without random oracles,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 3152, pp. 443–459, 2004.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5452463, 8 pages
https://doi.org/10.1155/2018/5452463

Research Article

Secure Storage and Retrieval of IoT Data Based on Private Information Retrieval

Khaled Riad [1,2] **and Lishan Ke** [3]

[1] *School of Computer Science, Guangzhou University, Guangzhou 510006, China*
[2] *Mathematics Department, Faculty of Science, Zagazig University, Zagazig 44519, Egypt*
[3] *College of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Khaled Riad; khaled.riad@science.zu.edu.eg and Lishan Ke; kelishan@gzhu.edu.cn

Received 25 May 2018; Accepted 29 August 2018; Published 18 November 2018

Guest Editor: Karl Andersson

The fast growth of Internet-of-Things (IoT) strategies has actually presented the generation of huge quantities of information. There should exist a method to conveniently gather, save, refine, and also provide such information. On the other hand, IoT data is sensitive and private information; it must not be available to potential attackers. We propose a robust scheme to guarantee both secure IoT data storage and retrieval from the untrusted cloud servers. The proposed scheme is based on Private Information Retrieval (PIR). It stores the data onto different servers and retrieves the requested data slice without disclosing its identity. In our scheme, the information is encrypted before sending to the cloud servers. It is also divided into slices of a specific size class. The experimental analysis on many different configurations supported efficiency and the efficacy of the proposed scheme, which demonstrated compatibility and exceptional performance.

1. Introduction

With the huge revolution of Internet-of-Things (IoT) and cloud computing as its storage environment, the user requests a query to a part of information and should receive that part without disclosing its identity. A great number of researches have been dedicated to defend the database from *curious* users. There are approaches that enable questions to be asked by an individual into a database by reconstructing the worth of entities in a manner that prevents him. If the user would like to maintain his privacy (in the information-theoretic sense), then he could request a copy of the entire database. This can cause a huge communication overhead, making it unacceptable.

Before going further let us make the problem more tangible. Let a binary string $x = x_1, \ldots, x_n$ of length n. $k \geq 2$ server stores copies of this binary string. The user has some indicator $i \leq n$ and he is interested in getting this little x_i. To attain this aim, the little x_i could be calculated, the user queries each of the servers and receives responses. The query to each server is distributed separately of i and each server gains no information about i. A strategy with these properties is called a Private Information Retrieval (PIR) [1, 2].

Within this paper we introduce encrypted PIR that offers a great privacy. That is, unbounded servers should not obtain any information about the requested piece of information. One does require at least two servers to achieve privacy. These servers do not need to store the whole database; they could store portions of it. We show that when those pieces are encrypted, instead of duplicated, the memory overhead can be decreased.

1.1. Motivation. In addition to reducing the storage overhead caused by replicating the data to reduce the communication cost in the traditional PIR protocols [2] and achieving the information-theoretic privacy, in big data, the user who reconstructs data is distinct from the user who distributes them. Also, the user who distributes data should encrypt it using different keys and distribute the ciphertext.

Moreover, querying information over big data where no one can get the identity of the parts you are querying or the responses obtained is a big challenge and sounds like science fiction. But it is actually the PIR science. In modern data storage systems, data are usually stored at multiple storage nodes in the cloud. The privacy of information retrieval has to be shielded. One naive method to attain PIR is by simply downloading each document in the system regardless of the user requirements. The drawback of that strategy is the very large restoration cost, which further increases with N (the range of saved documents). Thus, there is an urgent requirement to present a strong PIR strategy for storage and recovery.

1.2. Contributions. One of the main contributions in this paper is integrating PIR [1] with cloud computing, to guarantee the efficiency and security of encrypted storage and retrieval of information for big data queries from the untrusted cloud servers. Our scheme first divides the data into slices, then encrypts each of them, and stores those encrypted slices using the Swift service on the untrusted cloud servers. In the reconstruction stage, the requested data slice is reconstructed without disclosing any information about that requested slice, with retrieval cost independent of the number of stored slices. The main contributions can be summarized as follows:

(i) We have integrated the PIR scheme with cloud computing, to secuerly store the personal information as well as effiently and smothly reterive it.

(ii) We have implemented our scheme on the top of our private cloud environment, by initializing a set of virtual instances that are divided into three categories based on their configuration and their role in the proposed scheme.

(iii) We have tested the proposed scheme under two different scenarios, client situation and overlay situation.

(iv) The experimental results have suggested that the decryption and retrieval cost are separate from the amount of saved slices and harmonious with all reasonable scenarios for applying our proposed scheme.

Ultimately, the throughput for a workload caused by the implementation of mixing get and also put_fragment requests on Swift is sensible and accepted by the operator and user.

1.3. Organization. The rest of this paper is organized as follows: Section 2 presents the related work and smooth comparison between the currently proposed information hiding and extraction approaches while not disclosing the queried information. Section 3 presents our proposed *k-private and t-out-of-n PIR* scheme and its three stages. The proposed scheme implementation is introduced in Section 4. The comprehensive performance analysis is presented in Section 5. This is followed by the conclusion in Section 6.

2. Related Work and Discussion

The notion of earning the extraction of this data content of a huge data source should consider quite rough security requirements to be able to do not disclose the queried data. Reconstructing a part of discerning information from an encrypted source is determined by the availability of the whole source, which is introduced by Rivest [3], by proposing *all-or-nothing transform* (AONT). In this scheme any modification on the encrypted message limits the ability to decrypt the resource. Thus, the AONT scheme works well for the scenarios where the user who wants to decrypt the resource has never accessed the key before. This is not realistic, because the cloud users frequently access the resources and request to decrypt their ciphertext. In our scheme, the user can decrypt the resource as long as he has the appropriate key and the sufficient data slices that can generate the requested resource. Moreover, the requesting user can only decrypt the ciphertext if he has successfully generated the decryption token. Another direction for securely uploading the data to cloud is introduced in [4], which ensures that the cloud validates the data integrity while avoiding malicious home gateways that monitor and modify the data. In our scheme the data is not stored as one block, but it is sliced into several slices based on its size. Then, it is first hidden using a permutation hash function. After that, those data slices are encrypted using and encryption algorithm and an access structure that is implicitly included in the ciphertext. In this manner we are reducing the storage time overhead. The authors in [5] formalize the notion of verifiable database with incremental updates (Inc-VDB). As well as in [6] pointing out to Catalano-Fiore's VDB framework from vector commitment is vulnerable to the so-called forward automatic update (FAU) attack.

For information hiding and securely extracting that information, there are multiple contributions based on different schemes, such as [7]. The authors introduced a T-private PIR which is a generalization of PIR to include the requirement that even if any T of the N databases colludes, the identity of the retrieved message remains completely unknown to them. But in our scheme whatever was the number of colluding databases, the user will only receive the encrypted data slices that are only sufficient for extracting the required information not more. Also, the authors in [8] utilized a new cryptographic primitive, called conditional disclosure of secrets, which we believe that it may be a useful building block for the design of other cryptographic protocols. In our scheme, we have considered slicing the data into multiple slices before encrypting and storing it on the cloud storage servers. The authors in [9] proposed a new symmetric encryption for mobile devices. The authors in [10] have introduced a new operative scheme called RoughDorid for detecting malware applications directly on the smartphone. A Dynamic Fully Homomorphic Encryption-based Merkle Tree is proposed in [11]. An outsourced revocation has been introduced in [12] based on identity-based encryption in cloud computing. A new privacy preserving response scheme has been proposed in [13] based on adaptive key evolution in smart grid. A novel dynamic structure for cloud data has been proposed in [14] for efficient public auditing.

Other strategies for applying access control in the cloud via encryption are developed together two research lines: attribute-based encryption (ABE) and discerning encryption procedures. ABE approaches (e.g., [15–18]) supply access control authorities by making sure that the key used to protect

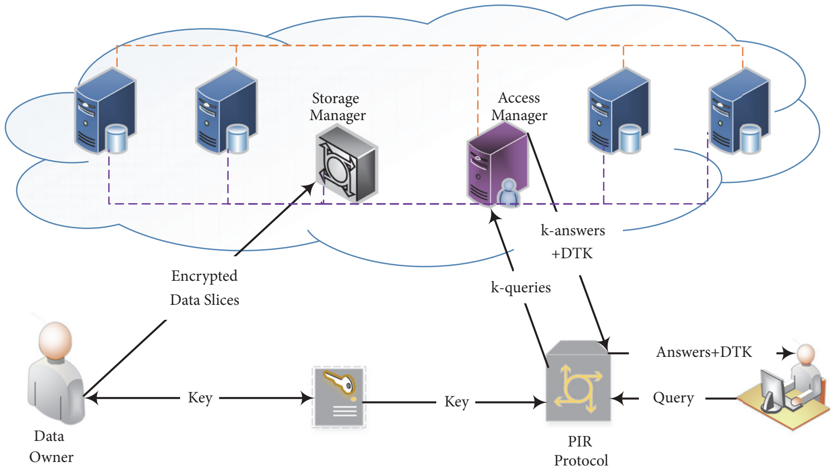


Figure 1: Integrated PIR with cloud computing system model.

a source could be derived exclusively by the consumers that meet a specified condition in their characteristics (e.g., era, function). An efficient and secure data outsourcing with check-ability has been proposed in [19] for cloud computing based on ABE. Also, a novel lightweight encryption mechanism for database is introduced in [20]. The authors in [21] introduced a new identity-based signcryption on lattice without trapdoor. For the multiple sources, the authors in [22] proposed a new homomorphic signature scheme based on network coding and applied it to IoT. The privacy is preserved in IoT using centralized duplicate removal video storage system [23]. Also, a new identity-based antiquantum blind authentication for privacy preserving in wireless sensor networks has been proposed in [24]. For the blind storage, the authors in [25] have introduced efficient multikeyword ranked search for mobile cloud data. In [26] the authors proposed the personalized search in mobile clouds over encrypted data with efficient updates. Reference [27] proposed a new block design-based key agreement for data sharing in cloud computing.

Procedures based on discerning encryption (e.g., [28, 29]) suppose to encrypt every single source using a secret that only licensed users understand or may derive. Within this situation, the information owner either manages policy upgrades, with overhead, or is assigned to the server. Though overencryption is shown to provide functionality that is decent and promises a prompt authorities of policy upgrades, it needs trust premises that are more powerful on the machine, which has to offer support. Also, another set of contributions [30, 31] have implemented access control for a private cloud computing evironment, by proposing the confidence notation for denying or granting access. In the event the host is oblivious of its adoption our scheme may be used. The authors in [32] proposed a flexible EHR sharing scheme supporting offline encryption of EHR and outsourced decryption of EHR ciphertexts in mobile cloud computing. Reference [33] proposed the first lattice-based linearly homomorphic signature in the standard model, which settles this open problem. The authors in [34] proposed a dependable distributed WSN framework for SHM. Then the authors in [35] proposed a new biometrics-based authentication scheme for the multiserver environment.

3. PIR with Cloud Computing

The challenge yet is the way to look for an efficient and protected PIR scheme (regarding costs for information storage and recovery). Our proposed scheme is *k-private and t-out-of-n PIR*, which means that if only t of n servers is required to respond and even though k servers collude together, the queried information will not be revealed. Our scheme consists of three basic stages: *Data Storage and Encryption*, *User Authorization and Query*, and *Data Decryption and Reconstruction*. We have employed the basic PIR scheme definition [36].

3.1. Data Storage and Encryption

3.1. Data Storage and Encryption. The data $\mathscr{D}$ will be stored on k cloud server $(\mathscr{SRV}_1, \mathscr{SRV}_2, \dots, \mathscr{SRV}_i, \dots, \mathscr{SRV}_k,)$ after encrypting it with our encryption algorithm. We consider three types of servers in the data storage side, as shown in our system model Figure 1. The *Storage Manager Sever* receives the encrypted data slices and distributes them on the *Data Storage Servers* under its own control, and the *Key-Server* is responsible for the key management that is distributed using the *PIR Protocol*.

The owner encrypts all the information bits using a content key using symmetric encryption procedures. The content essential is then encrypted by the owner. It requires as inputs the public key handled by the *PIR Protocol*, the content key *ck*, and an access structure $\mathscr{A} = (M, \rho)$. Let M be a $l \times n$ matrix, where l denotes the total number of all the attributes. The function ρ associates rows of M to attributes.

$$\mathscr{D} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_i^1 & \cdots & x_k^1 \\ x_1^2 & x_2^2 & \cdots & x_i^2 & \cdots & x_k^2 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ x_1^j & x_2^j & \cdots & x_i^j & \cdots & x_k^j \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ x_1^n & x_2^n & \cdots & x_i^n & \cdots & x_k^n \end{bmatrix}$$

$$= \begin{bmatrix} \overline{x}_1^{h(1)} & \overline{x}_2^{h(1)} & \cdots & \overline{x}_i^{h(1)} & \cdots & \overline{x}_k^{h(1)} \\ \overline{x}_1^{h(2)} & \overline{x}_2^{h(2)} & \cdots & \overline{x}_i^{h(2)} & \cdots & \overline{x}_k^{h(2)} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \overline{x}_1^{h(j)} & \overline{x}_2^{h(j)} & \cdots & \overline{x}_i^{h(j)} & \cdots & \overline{x}_k^{h(j)} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \overline{x}_1^{h(n)} & \overline{x}_2^{h(n)} & \cdots & \overline{x}_i^{h(n)} & \cdots & \overline{x}_k^{h(n)} \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & \cdots & s_i & \cdots & s_k \end{bmatrix} \tag{1}$$

where each element of the vector $[s_1, s_2, \dots, s_i, \dots, s_k]$ represents the data to be stored on each server $(\mathcal{SRV}_1, \mathcal{SRV}_2, \dots, \mathcal{SRV}_i, \dots, \mathcal{SRV}_k,)$ respectively. Each s_i is the corresponding column after applying $h(j)$ on each piece of data. $h(j)$ is our hash permutation function that is used to hide the index of each piece of the stored data. More precisely, a hash function h maps bit strings of arbitrary finite length to strings of fixed length, say n bits. For a domain D and range R with $h : D \longrightarrow R$ and $|D| > |R|$, the function is many-to-one, implying that the existence of collisions (pairs of inputs with identical output) is unavoidable.

Definition 1 (Hash Permutation Function h). A hash function is a function $h : D \longrightarrow R$ and $|D| > |R|$ maps bit strings of arbitrary finite length to strings of fixed length, which has the following properties:

(1) Compression: h maps an input x_i of arbitrary finite bit length, to an output $h(x_i)$ of fixed bit length n.

(2) Ease of computation: given h and an input x_i, $h(x_i)$ is easy to compute.

(3) Preimage resistance: for essentially all prespecified outputs, it is computationally infeasible to find any input which hashes to that output, i.e., to find any preimage x_i such that $h(x_i) = y$ when given any y for which a corresponding input is not known.

(4) 2nd-preimage resistance: it is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x_j, to find a 2nd-preimage $x_i \neq x_j$ such that $h(x_i) = h(x_j)$.

(5) Collision resistance: it is computationally infeasible to find any two distinct inputs x_i, x_j which hash to the same output, i.e., such that $h(x_i) = h(x_j)$.

We can find that the collision resistance implies 2nd-preimage resistance of hash functions, but collision resistance does not guarantee preimage resistance.

3.2. User Authorization and Query. The user can issue a query $\mathcal{Q}$ to receive a file x which is partitioned and stored on k different server $(\mathcal{SRV}_1, \mathcal{SRV}_2, \dots, \mathcal{SRV}_i, \dots, \mathcal{SRV}_k,)$. Considering that the information is hosted on both cloud servers. Then there has to be a decryption token for every user to have the ability to synchronize the information. The user has to issue a query to the PIR protocol including the user's secret key, attributes, and certificates. The PIR protocol will issue k-queries (one for each server) $\mathcal{Q} = \{q_1, q_2, \dots, q_i, \dots, q_k\}$ for the access manager server, where $q_i = \mathcal{Q}_i(k, n, j)$ and j is a randomly chosen by flipping coins. Each server will respond with a single encrypted answer; the user will have an encrypted answer set $EA = \{ea_1, ea_2, \dots, ea_i, \dots, ea_k\}$, where $ea_i = EA_i(k, i, x, q_i)$. The access manager server will reply with the servers' answers and the decryption token that is sent to the user. Then, it will have the ability to decrypt and rebuild the requested information.

3.2.1. Decryption Token Generation. The decryption token generation algorithm (Algorithm 2) is run by the access manager server. It requires as inputs the ciphertext CT that implicitly includes an access structure $\mathcal{A}$, user's public key UPK generated by the PIR protocol, user's secret key USK, and user's set of attributes USA. When USA suits the accessibility construction $\mathcal{A}$, the algorithm could successfully calculate the right decryption token DTK of this ciphertext. Thus, the PIR protocol can transform the user's query to a set of k-queries for the *Access Manager*.

3.3. Data Decryption and Reconstruction. Once the user has received the encrypted answers set EA and its decryption token DTK, he can decrypt the answers with the help of its own secret key USK and get the answer set A that will be used for reconstruction. Since the permutation function can make some collusions (NC) probability p, the success probability is $1 - (1 - p)^k$, and the reconstruction threshold $t = k - NC$. Based on the value of t the user will be able to reconstruct the requested slice. Once the data has been reconstructed, the user can use its given decryption token to decrypt the data.

4. Implementation

Our model is implemented on the top of our private cloud environment, which is based OpenStack [37]. We have built our proposed scheme by initializing a set of virtual instances that are divided into three categories based on their configuration:

(i) **Category 1:** n virtual instances working as storage servers. The configuration of those n servers is 4 VCPUS, 4 GB RAM, and 80 GB disk. We have considered $n = 50$; thus, the IP addresses for those servers are 10.0.10.101 : 10.0.10.150 with subnet mask 255.255.255.0. Those storage servers have two basic tasks, storing the encrypted data slices submitted to them through the *Storage Manager*. The second task is sending the encrypted data slices back as answers to the *Access Manager* server to be delivered to the requesting user.

(ii) **Category 2:** two virtual instances. The configuration for each of them is 4 VCPUS, 8 GB RAM, and 20 GB disk. They are working as *PIR Protocol* server with IP address 10.0.10.151 and *Key Manger* server with IP address 10.0.10.152. After the k *– answers* are received by the *PIR Protocol* from the *Access Manager*

```
Input: (i) $ck_i$                                  ▷ The content key for each $x_i$
       (ii) $PK$   ▷ The data $\mathscr{D}$ public key managed by Shamir secret sharing
       (iii) $\mathscr{A} = (M, \rho)$            ▷ The data $\mathscr{D}$ access structure, which will be
           implicitly included in the ciphertext
Choose $\alpha_i, \beta_i, \gamma_i \in \mathbb{Z}_q^*$          ▷ The random secrets of each data slice $x_i$
Choose $f \in \mathbb{Z}_q^*$                          ▷ A random encryption exponent
Choose $\overrightarrow{v} = [f, y_2, \ldots, y_n]^T \in \mathbb{Z}_q^{*n}$   ▷ A random vector, where $[y_2, \ldots, y_n]$
  are used to share the random encryption exponent $f$
for $j = 1$ to $l$ do
    Compute $\lambda_j = \overrightarrow{v}.M_j$ ▷ $M_j$ is the vector corresponding to the j-th row of
  the matrix $M$
end for
Randomly choose $r_1, \ldots, r_l \in \mathbb{Z}_q^*$
Compute: $C' = g^f$ and $C'' = g^{f/\beta_i}$
for $i = 1$ to $l$ do
    Compute $C_j = g^{a\lambda_j}.((g^{\overrightarrow{v}(j)\rho(j)}H(\rho(j)))^{\gamma_i})^{-r_j}$
    Compute $D_{1,j} = g^{r_j/\beta_i}$ and $D_{2,j} = g^{-(\gamma_i/\beta_i)r_i}$
end for
Compute the ciphertext:
    $CT_i = \left[ck_i.\left(\prod_{j}^{l}\widehat{e}(g,g)^{\alpha_i}\right)^{f}, C', C'', C_j, D_{1,j}, D_{2,j}, \rho(j)\right]$
Output: The ciphertext $CT$
```

Algorithm 1: Encryption.

server, the user must not be given so much data slices to guarantee the data secrecy. Also, the user should not be given less data slices than the required slices that can generate the requested data. Thus, the *PIR Protocol* server runs the PIR protocol to ensure hiding the requested data slices identity from the cloud storage servers and also ensuring granting the user the appropriate data slices to be able to recover the requested data. The *Key Manager* server is responsible for assigning the keys for both the data owners and requesting users.

(iii) **Category 3:** two virtual instances. The configuration for each of them is 8 VCPUS, 16 GB RAM, and 40 GB disk. They are working as *Storage Manager* server with IP address 10.0.10.154 and *Access Manager* server with IP address 10.0.10.155. The *Storage Manager* receives the encrypted data slices and distributes them on the appropriate cloud storage instances. The *Access Manager* receives the encrypted data slices from the cloud storage instances and sends them to the *PIR Protocol*.

It should be mentioned that all of those virtual instances are cooperating together to construct the proposed scheme. Each of them has its own task that can perfectly execute it.

5. Performance Analysis

The performance analysis of our proposed scheme is introduced based on two mandatory scenarios.

5.1. Client Situation. Our scheme requires the user to perform a more intricate decryption compared to using Attribute Encryption Scheme (AES) using a conventional encryption manner, by introducing the Token Generation Algorithm 2. In our scheme the decryption is parallelized on several core VCPUs, which makes the user processing much more effective. In our experiments, we have considered five different size categories: *32:127 bits; 128:511 bits; 512:2047 bits; 2048:8191 bits; and 8192:32768 bits*.

Figure 2 shows that the decryption cost can be used with all acceptable scenarios for the use of our scheme. Specifically, the figure illustrates the throughput obtained by changing the number of slices, through implementing our scheme in various configurations defined by five size groups *(32:127 bits, 128:511 bits, 512:2047 bits, 2048:8191 bits, and 8192:32768 bits)*. Based on the execution of our encryption protocol (Algorithm 1), we notice that even the largest size category *(8192:32768 bits)* offers a throughput that is approximately 85 MB/s, while considering 16 slices of that size category. The throughput for the smallest size category *(32:127 bits)* is about 140 MB/s, while considering 16 slices of that size category. The figure also reveals that decreasing the amount of slices, we achieve the performance level that is 1.5 times that obtained from the *8192:32768 bits* size group and 2 times the one obtained by *32:127 bits* size group. It should be noted that the experimental results for that part are the average of *25* trails.

5.2. Overlay Situation. In our proposed scheme, the overlay situation is analyzed by using the Swift (it organizes objects

```
Input:  (i) $CT_x$          ▷ The ciphertext related to the file $x$
        (ii) $UPK$          ▷ The user's public key given by PIR protocol
        (iii) $USK$         ▷ The user's secret key
        (iv) $UA$           ▷ The user's set of attributes
Let: $CT_A = |\mathscr{A}|$         ▷ The set of attributes involved in $CT_x$
Choose a set of constants $w_i \in \mathbb{Z}_q^*, \forall i \in CT_A$
for $i = 1$ to $CT_A$ do
    if $\lambda_i \in Share(f)$ then ▷ $\lambda_i$ are valid shares of the secret $f$ according to the data $\mathscr{D}$
        Reconstruct the encryption exponent: $f = \sum_{i=1}^{CT_A} w_i\lambda_i$
    end if
end for
Let $\{R_{U,i}, K_{U,i}, L_{U,i} \in \mathbb{Z}_q^*\}_{\forall i\in CT_A}$ are random constants
if $UA \models \mathscr{A}$ then            ▷ The user's attributes satisfies $\mathscr{A}$
    $DTK = \prod_{i=1}^{CT_A} \frac{\hat{e}(C', K_{U,i}).\hat{e}(C'', R_{U,i})^{-1}}{\prod_{i=1}^{CT_A} \left[\hat{e}(C_i, UPK_i).\hat{e}\left(D_{1,i}, K_{U,\rho(i)}\right).\hat{e}\left(D_{2,i}, L_{U,i}\right)\right]^{w_i CT_A}}$
        $= \frac{\hat{e}(g,g)^{a.U.f.CT_A} . \prod_{i=1}^{CT_A} \hat{e}(g,g)^{(\alpha_i/USK)f}}{\hat{e}(g,g)^{a.U.f.CT_A} . \sum_{i=1}^{CT_A} w_i\lambda_i}$
        $= \prod_{i=1}^{CT_A} \hat{e}(g,g)^{(\alpha_i/USK)f}$
else                                ▷ $DTK$ cannot successfully computed
    $DTK$ = rand($Hex$)             ▷ $DTK$ will get a random Hexadecimal value
end if
Output: The Decryption Token $DTK$
```

Algorithm 2: Token generation.

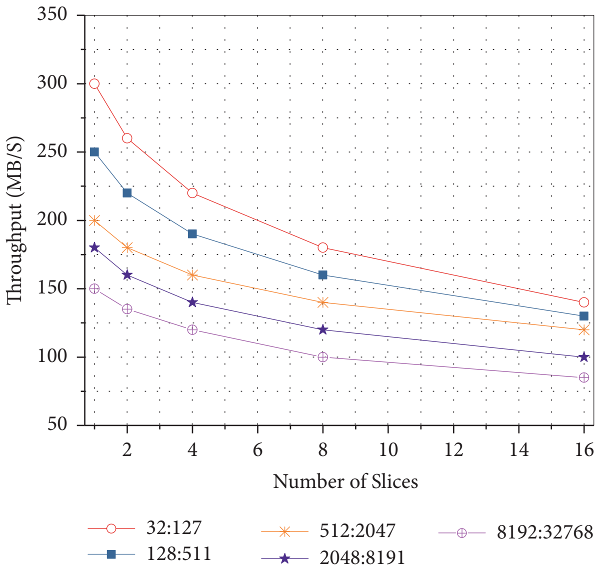


Figure 2: Throughput varying the number of considered slices for four different size categories.

within containers) service as a reference. We have adopted the DLO support offered by Swift to implement the get and put_fragment methods that characterize our scheme. With our experiments at the overlay scenario, we have assembled a Swift user program in Python that implements the get and also put_fragment techniques that describe our strategy. We have followed two strategies: one is by fragmenting an object as atomic objects; the second is by using DLO introduced by Swift.

Figure 3 contrasts the period required for the implementation of get requests based on different amounts of contemplated fragments *(1, 4, 16, 64, 256, and 1024)* to a specific object. The operation is droved dependent on the network bandwidth and also the overhead imposed by the management of every get request.

Specifically for get requests, the overhead introduced into handling one portion for every fragment predominates in the event of little costs, whereas the growth in object's size exhausts the system's bandwidth that causes a great bottleneck. By contemplating an item of size 1 GB, the time required for implementing the get requests is roughly 92 seconds for 1 considered fragment, and the time is 1000 seconds for 1024 considered fragments for the same object size. In case of considering an object of size 64 KB, the time required for executing the get requests is about 0.08 seconds for 1 considered fragment, and the time is 50 seconds for 1024 considered fragments for the same object size.

Figure 4 reports the throughput for a workload caused by the implementation of blending get and also put_fragment requests on Swift by changing the object size and the amount

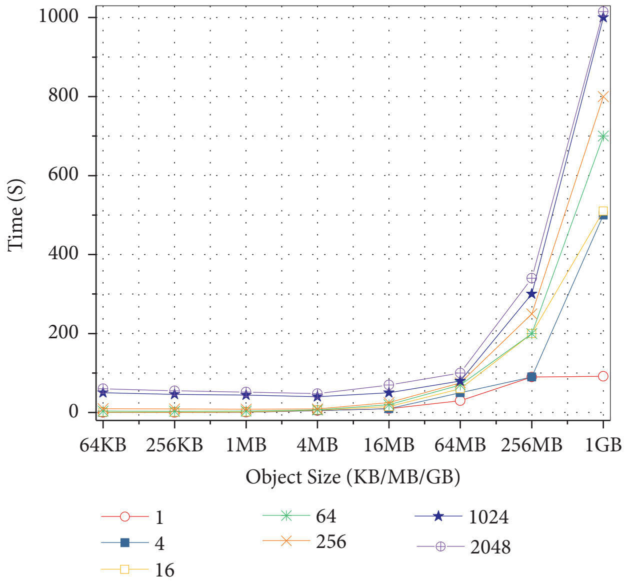


FIGURE 3: The execution time for the get requests on Swift.

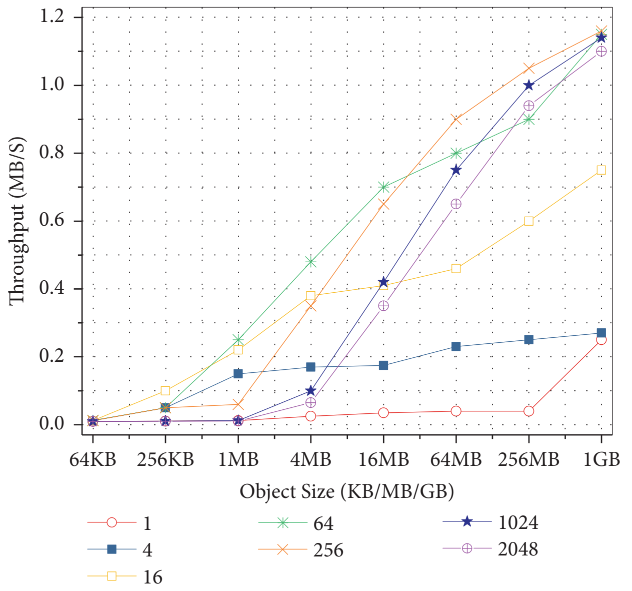


FIGURE 4: The throughput for a workload introduced by combining the get and put_fragment requests on Swift.

of contemplated fragments for the exact same object. We have assessed the behavior of our scheme based on a selection of 2048 objects, where following every put_fragment request, a succession of 100 get requests were implemented on objects in precisely the exact same group and are all of the exact same size. These configurations using fragments' throughput will be orders of magnitude greater already. The figure also demonstrates that the very best number of fragments is dependent upon the resource size. The identification of this value needs to think about the setup of the workload and this machine.

6. Conclusion

We presented a robust PIR scheme for efficiently and securely storing and retrieving private information from untrusted cloud servers. Our scheme lets the data owners to effectively divide and encrypt their own data into small slices of five different size categories. Our implementation and experimental analysis confirm the efficacy and efficacy of our proposed scheme, which appreciates orders of magnitude of improvement in throughput concerning source protection and decryption. For an object of size 1 GB, the time required for implementing the get requests is roughly 92 seconds for 1 considered fragment; the time is 1000 seconds for 1024 considered fragments for the same object size. Considering an object of size 64 KB, the time required for executing the get requests is about 0.08 seconds for 1 considered fragment, and the time is 50 seconds for 1024 considered fragments for the same object size. The proposed scheme also supports its compatibility with all present cloud storage environments, which makes it also relevant to a lot of application domains.

Data Availability

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

Conflicts of Interest

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

References

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proceedings of the 36th Annual Foundations of Computer Science*, pp. 41–50, Los Alamitos, Calif, USA, October 1995.

[2] B. Chor and N. Gilboa, "Computationally private information retrieval (extended abstract)," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pp. 304–313, New York, NY, USA, May 1997.

[3] R. L. Rivest, "All-or-nothing encryption and the package transform," in *Fast Software Encryption*, vol. 1267 of *Lecture Notes in Computer Science*, pp. 210–218, Springer Berlin Heidelberg, 1997.

[4] J. Shen, C. Wang, T. Li, X. Chen, X. Huang, and Z.-H. Zhan, "Secure data uploading scheme for a smart home system," *Information Sciences*, vol. 453, pp. 186–197, 2018.

[5] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 10, pp. 3184–3195, 2016.

[6] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New Publicly Verifiable Databases with Efficient Updates," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 546–556, 2015.

[7] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 64, no. 4, part 1, pp. 2361–2370, 2018.

[8] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," *Journal of Computer and System Sciences*, vol. 60, no. 3, pp. 592–629, 2000.

[9] C. Gao, S. Lv, Y. Wei, Z. Wang, Z. Liu, and X. Cheng, "M-SSE: an effective searchable symmetric encryption with enhanced security for mobile devices," *IEEE Access*, vol. 6, pp. 38860–38869, 2018.

[10] K. Riad and L. Ke, "Operative scheme for functional android malware detection," *Security and Communication Networks*.

[11] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C. Gao, "Dynamic Fully Homomorphic encryption-based Merkle Tree for lightweight streaming authenticated data structures," *Journal of Network and Computer Applications*, vol. 107, pp. 113–124, 2018.

[12] J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 425–437, 2015.

[13] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2014.

[14] 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, no. 10, pp. 2402–2415, 2017.

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

[16] K. Riad, "Multi-authority trust access control for cloud storage," in *Proceedings of the 4th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS '16)*, pp. 429–433, August 2016.

[17] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[18] K. Riad, "Revocation basis and proofs access control for cloud storage multi-authority systems," in *Proceedings of the 3rd International Conference on Artificial Intelligence and Pattern Recognition (AIPR '16)*, pp. 118–127, September 2016.

[19] J. Li, X. Y. Huang, J. W. Li, X. F. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2201–2210, 2014.

[20] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, and D. S. Wong, "L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing," *Knowledge-Based Systems*, vol. 79, pp. 18–26, 2015.

[21] X. W. Y. Zhang, H. Zhu, and L. Jiang, "An identity-based signcryption on lattice without trapdoor," *Journal of Universal Computer Science*, 2018.

[22] T. Li, W. Chen, Y. Tang, and H. Yan, "A homomorphic network coding signature scheme for multiple sources and its application in IoT," *Security and Communication Networks*, vol. 2018, Article ID 9641273, 6 pages, 2018.

[23] H. Yan, X. Li, Y. Wang, and C. Jia, "Centralized duplicate removal video storage system with privacy preservation in IoT," *Sensors*, vol. 18, no. 6, 2018.

[24] H. Zhu, Y. Tan, L. Zhu, X. Wang, Q. Zhang, and Y. Li, "An identity-based anti-quantum privacy-preserving blind authentication in wireless sensor networks," *Sensors*, vol. 18, no. 5, 2018.

[25] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 1, pp. 127–139, 2015.

[26] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 97–109, 2018.

[27] 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*, 2018.

[28] S. D. C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Encryption policies for regulating access to outsourced data," *ACM Transactions on Database Systems (TODS)*, vol. 35, no. 2, 2010.

[29] I. Hang, F. Kerschbaum, and E. Damiani, "ENKI: Access control for encrypted query processing," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '15)*, pp. 183–196, New York, NY, USA, June 2015.

[30] K. Riad, "Blacklisting and forgiving coarse-grained access control for cloud computing," *International Journal of Security and Its Applications*, vol. 10, no. 11, pp. 187–200, 2016.

[31] K. Riad and Z. Yan, "Multi-factor synthesis decision-making for trust-based access control on cloud," *International Journal of Cooperative Information Systems*, vol. 26, no. 04, pp. 1–33, 2017.

[32] Z. Cai, H. Yan, P. Li, Z.-A. Huang, and C. Gao, "Towards secure and flexible EHR sharing in mobile health cloud under static assumptions," *Cluster Computing*, vol. 20, no. 3, pp. 2415–2422, 2017.

[33] W. Chen, H. Lei, and K. Qi, "Lattice-based linearly homomorphic signatures in the standard model," *Theoretical Computer Science*, vol. 634, pp. 47–54, 2016.

[34] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable structural health monitoring using wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 363–376, 2017.

[35] H. Shen, C. Gao, D. He, and L. Wu, "New biometrics-based authentication scheme for multi-server environment in critical systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 6, pp. 825–834, 2015.

[36] A. Fazeli, A. Vardy, and E. Yaakobi, "PIR with Low Storage Overhead: Coding Instead of Replication," 2015, https://arxiv.org/abs/1505.06241.

[37] OpenStack, http://www.openstack.org/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7584942, 8 pages
https://doi.org/10.1155/2018/7584942

Research Article

Multiresolution Face Recognition through Virtual Faces Generation Using a Single Image for One Person

Hae-Min Moon,[1] Min-Gu Kim,[2] Ju-Hyun Shin ,[3] and Sung Bum Pan [4]

[1] *Management & Planning Division, Korea Invention Promotion Association, 621-15, Docheun-dong, Gwangsan-gu, Gwangju, Republic of Korea*
[2] *Department of Control and Instrumentation Engineering, Chosun University, 375 Seosuk-dong, Dong-gu, Gwangju 61452, Republic of Korea*
[3] *Department of ICT Convergence, Chosun University, 375 Seosuk-dong, Dong-gu, Gwangju 61452, Republic of Korea*
[4] *Department of Electronics Engineering, Chosun University, 375 Seosuk-dong, Dong-gu, Gwangju 61452, Republic of Korea*

Correspondence should be addressed to Sung Bum Pan; sbpan@chosun.ac.kr

Received 2 July 2018; Revised 27 September 2018; Accepted 23 October 2018; Published 11 November 2018

Academic Editor: Ilsun You

In recent years, various studies have been conducted to provide a real-time service based on face recognition in Internet of things environments such as in a smart home environment. In particular, face recognition in a network-based surveillance camera environment can significantly change the performance or utilization of face recognition technology because the size of image information to be transmitted varies depending on the communication capabilities. In this paper, we propose a multiresolution face recognition method that uses virtual facial images by distance as learning to solve the problem of low recognition rate caused by communication, camera, and distance change. Face images for each virtual distance are generated through clarity and image degradation for each resolution, using a single high-resolution face image. The proposed method achieved a performance that was 5.9% more accurate than methods using MPCA and SVM, when LDA and the Euclidean distance were employed for a DB that was configured using faces that were acquired from the real environments of five different streets.

1. Introduction

IoT is an intelligent system that helps communicate with people and things or between things and things using Internet networks. With the recent advancement in various technologies such as high-speed networks, large-capacity data transmission, wired-wireless sensor networks, among others, not only IoT but also related application technologies have been actively developed. With the recent developments in hardware technology, a video surveillance system using a high-resolution camera is commonly used. However, in the case of an intelligent surveillance system, there is a limit to data transmission due to high computational real-time analysis. Therefore, thanks to the spread of high-speed communication technology, such as 5G, the high-resolution-based surveillance camera has been applied to remote face recognition technology. Along with the recent increase in the availability of high-resolution cameras, there has also been an increase in the use of image monitoring systems based on such cameras to improve face recognition performance. In face recognition systems, faces are recognized using high-resolution face images that are acquired at close range, and these serve as an important factor that can guarantee an effective recognition performance. However, aside from access control systems, it is difficult for face recognition systems to acquire high-resolution face images at close range because the distance between the camera and the person varies. Images that are captured through the camera inevitably suffer in terms of the face resolution if the distance between the camera and the person becomes too wide, regardless of the cameras performance. In other words, even if high-resolution cameras are employed in face recognition systems, the face recognition performance cannot be guaranteed for long range low-resolution images [1].

Many studies exist regarding various face recognition methods, but issues still remain, such as the recovery

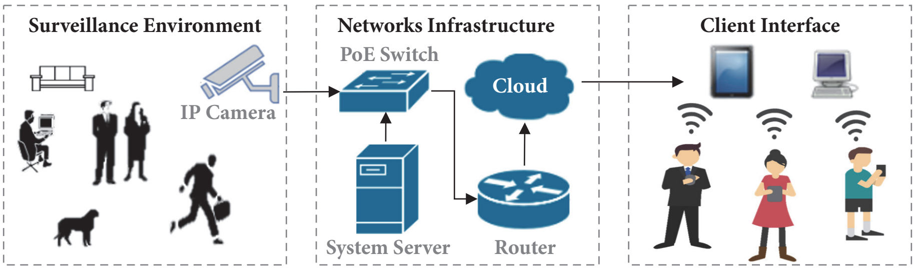


FIGURE 1: Structure of network-based surveillance system combined with IoT technology.

processing speed for low-resolution face images, reliance on learning data, and reduced recognition rates according to changes in distance that occur during actual face recognition situations. Although some studies are limited to a single environment, such as high or low-resolution face environments, no existing face recognition studies are based on environments that include changes in resolution. In other words, there is a need for face recognition technology based on multiresolution that is robust to changes in resolution resulting from camera performance or changes in distance [2]. This paper will propose a face recognition system that employs face recognition learning according to virtual resolutions in order to resolve the issue of reduced recognition rates resulting from changes in face image resolution that occur through changes in distance. Face images for each virtual distance are generated through clarity and image degradation for each resolution, using a single high-resolution face image. Clarity is measured using blur metrics, and Gaussian blurring is applied for image degradation. The performance of the proposed method is analyzed through a face DB that was configured using five different real resolutions. According to the test results, the proposed method achieved a performance that was 5.9% better than methods using MPCA and SVM, when LDA and the Euclidean distance were employed. When the standard image size was set to the average size of all face images at 30×30, under the conditions using LDA and the Euclidean distance, the average performance was improved by 40.7% for 16×16, 12×12, and 10×10, which are low-resolutions.

This paper is organized as follows. Section 2 explains related works with structure of network-based surveillance system. Section 3 explains face recognition using the proposed virtual multiresolution face images, Section 4 analyzes the experimental results, and Section 5 presents conclusions.

2. Related Work

In an IoT environment, the network camera-based facial recognition depends heavily on the camera because the recognition performance varies depending on the obtained image quality. Recently, UHD-class high-resolution cameras have become commonplace thanks to the rapid advancements in camera performance. However, in the case of network camera-based surveillance systems, there is a limit to its practical use due to the limit of transmission speed. Therefore, in this paper, we propose a facial recognition technology applicable to various resolutions. Figure 1 shows the structure of a network-based surveillance system using intelligent surveillance system technology combined with IoT in the network environment. First, the images obtained from the camera are transmitted to the network-based surveillance system for event analysis. The surveillance system decides if an event occurs by comparing the image received from the camera with the image stored in the server. Finally, when an event occurs, an alarm is sent to the user, which enables them to recognize the situation, and the event image is stored on the network server.

2.1. Video Surveillance System. The video surveillance system refers to a method of transmitting the image information captured by the camera to a receiver and observing a specific area that the user wants to be based on the transmitted image information. The video surveillance system is widely used for various applications in the fields of security, industry, vehicle surveillance, and traffic management. Recently, the availability of the system has been growing sharply due to the development of communication technologies such as IoT technology, 4G, and 5G. In addition, the advancement of IT technology has transformed an analog environment based on VCR storage devices into a digital age that uses DVR-based image compression and digital transmission technology. It has evolved into an intelligent video security technology that combines IP-based networks using broadband networks and open protocols with automatic image analysis and recognition technology. CCTV cameras can be divided into dome, box, IR, PTZ, and IP cameras depending on their use and shape. Considering camera performance is as important as selecting the right camera in a video surveillance system, that is, the use of the surveillance system is determined depending on the quality of the image obtained through the camera, high-resolution and high-quality images contain considerable information, which is a very important factor in intelligent surveillance systems based on image processing. Although an image is captured from the same location, in the case of the HD-class image, the information of the object can be intuitively confirmed without additional image processing, yet in the case of normal image quality, it is often difficult for us to directly understand the information of the object. That is, the higher the image resolution is, the clearer and more detailed image we can see. However, the price is more expensive depending on the camera performance, and the size of the image data becomes larger, resulting in more processing time.

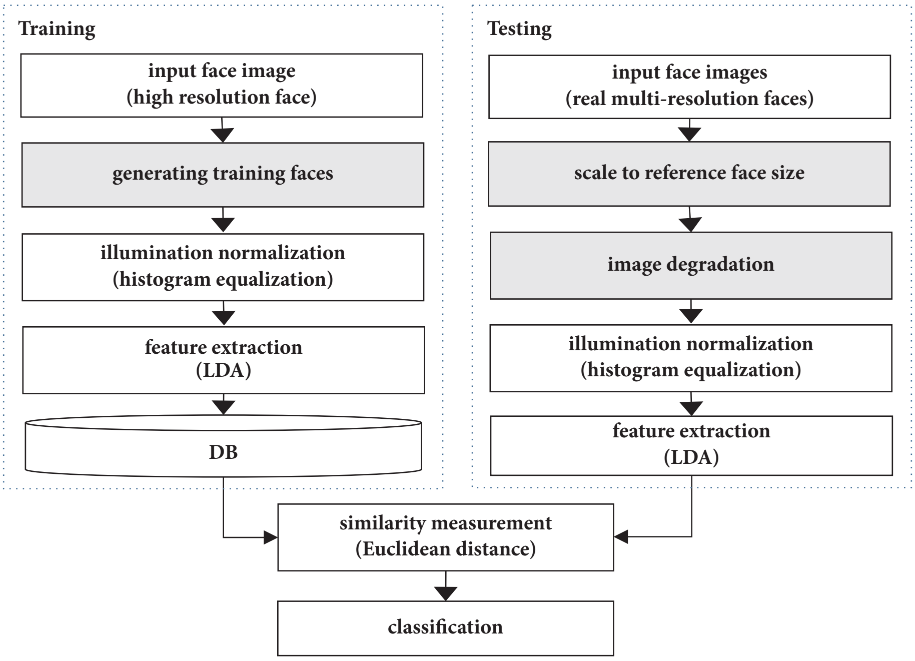


Figure 2: Flowchart of the proposed model for multiresolution face recognition.

2.2. Low-Resolution Based on Face Recognition Technology. Recently, studies have been conducted regarding face recognition technology using low-resolution images [3]. This face recognition method recognizes faces by using super resolution recovery, which converts low-resolution images into high-resolution images [4–6]. The super resolution recovery method achieves a superior high-resolution recovery performance from the visual end, but it is unable to achieve a satisfactory performance when it comes to face recognition. Because this method relies on learning data, it has the disadvantage of requiring high quantities of learning data, and processing times increase as the amount of data increases. There also exist studies on a method that exploits structural characteristics through the mapping and learning of a pair of low- and high-resolution images generated from the same face [3, 7, 8]. This method using structural characteristics is advantageous to the super resolution recovery method, owing to its lower calculational complexity. However, the recognition rate varies significantly according to undefined values such as changes in lighting, distances, face expressions, other external variables, and mapping related weighted values. Face recognition using a Pan-Tilt-Zoom(PTZ) camera boasts an extremely high face recognition rate because it can solve the fundamental issue of image deterioration from long range imaging [9, 10]. However, because the camera used in this method is extremely expensive, it cannot be employed for general purposes.

3. Proposed Algorithms for Multiresolution Face Recognition

Among face recognition methods that depend on learning, methods of improving the recognition rate include configuring data for testing and identifying the most similar faces through training data and increasing the amount of training data. The method of configuring testing data and identifying the most similar faces through learning uses face images that are extracted from various real resolutions. However, because this method acquires face images when the user is moving, it requires serious cooperation from users. This paper proposes a method that generates face images at various resolutions without requiring such cooperation from users.

Figure 2 illustrates the proposed face recognition method, where faces are recognized in multiresolution environments that were acquired by generating multiple low-resolution face images using a single high-resolution face image. Virtual low-resolution face images are scaled to a set reference face size. Bilinear interpolation is used for scaling, and histogram smoothing is applied to adjust the lightning. Faces for testing were not generated randomly, but rather five real resolution images were employed. At this point, image degradation was performed on the testing data by using face size normalization and imposing an image blur value. Real multiresolution faces for testing were acquired while varying the distance between the camera and the person between 1 m and 5 m. In order to recognize faces, LDA was used to extract features from the training and testing data, and the Euclidean distance was employed to measure the similarity [11, 12].

3.1. Generating Training Faces Using a Single Image per Person. In order to minimize the required calculations and use the least amount of samples, this paper proposes a method that employs a single high-resolution face image to automatically generate multiple face images for learning. Figure 3 illustrates the process of generating face images according to virtual

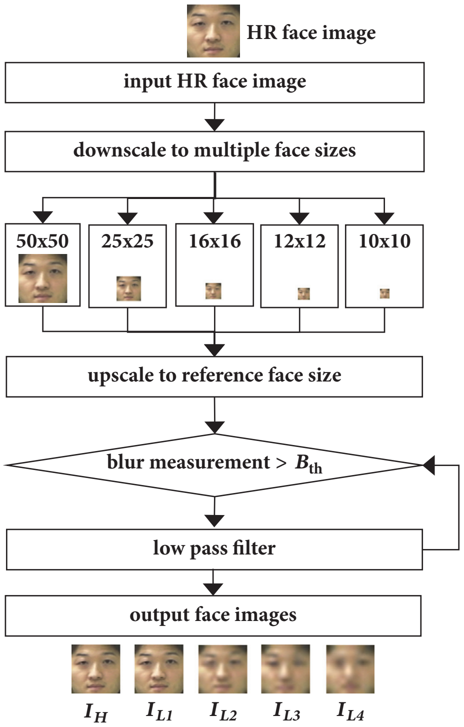


Figure 3: Flowchart of the proposed model for generating training faces using image degradation.

resolutions, using one high-resolution face image. One high-resolution face image is input and is then reduced to fit the standard size according to the resolution. The size of the original image that is reduced for each resolution is based on a real face image, which was detected as the distance between the person and the camera varied in 1 m intervals. The size of the original image in which faces can be detected is 320×240, and the distance between the person and the camera varies from 1 m to 5 m. Images that are reduced to the appropriate size according to the resolution are then expanded to be the same size as the reference image for face recognition. The definition of an expanded face image is reduced through low pass filtering until it reaches the target clarity. Equation (1) below defines the method for generating virtual face images using high-resolution face images.

$$\begin{gathered} I_{Li} = F_{int\,p}\left(RF_{size}, t^{G}_{Li}\right), \\ t^{G}_{Li} = \left(R_{ds}, I_{H}\right) \otimes G \end{gathered} \tag{1}$$

where I_{Li} is the face image at the *ith* virtual size that is used for learning, and RF_{size} is the expansion ratio for scaling to the standard image size for acquiring features.

Furthermore, t_{Li} represents the high-resolution image that is reduced to fit the face size according to each resolution. At this point, G indicates the blur kernel, ⊗ represents the convolution operation, and $F_{int\,p}$ represents the scaling. The high-resolution I_H is 50×50, while the low-resolution I_{L4} is 10×10. Finally, R_{ds} represents the scaling ratio for generating face images according to each resolution. The reduction ratio for face images at each resolution is acquired from the average face image size at each real resolution. Assuming that there are five face images for the training data for each candidate, $\Gamma_c = [I_H, I_{L1}, I_{L2}, I_{L3}, I_{L4}]$ is the group that is configured through the learning data for each candidate.

Figure 4 presents the results of comparing face images from each distance with virtual multiresolution face images. Figure 4(a) shows the high-resolution face image that was extracted from a distance of 1m, and Figure 4(b) shows the virtual face image that was generated using the proposed method. Figure 4(d) shows the face image for each resolution that was extracted from a real distance, and Figure 4(c) shows the image from Figure 4(d) after being normalized to fit the standard image size 30×30. The test results indicate that the virtual face image that was automatically generated was more similar to the real face image than the image produced from a set distance of 1m.

3.2. Image Degradation and Reference Face Size. Existing low-resolution face recognition technology employs a method that recovers face images to high-resolution to improve the face recognition rate. In this paper, the definition of a high-resolution face image is reduced through low pass filtering, which is the opposite of the existing method. The process of increasing the similarity through lowering the definition not only requires less calculation than existing high-resolution recovery technology but also does not require a reference image. In order to minimize the differences between the multiresolution training face image that is automatically generated and the real low-resolution resting face image, the clarity of the two images was considered.

The blur value of the image that is derived through the input, such as in (2), is measured and compared to the standard blur value for each resolution. If the blur value of the current input image is greater than the standard value B_{th}, then the definition is reduced through low pass filtering. This process is performed until the value becomes lower than the standard blur value.

$$t^{G}_{Li} = \begin{cases} t_{Li} \otimes if \ F_{blur}\left(t_{Li}\right) \geq B_{th} \\ t_{Li} \quad if \ F_{blur}\left(t_{Li}\right) < B_{th} \end{cases} \tag{2}$$

where F_{blur} represents the clarity measurement method, and the blur metric method [13], which does not employ a reference image, is used. G represents the blur kernel. The weighted value is generated through (3) and used to reduce the definition of the generated image.

$$G\left(x, y\right) = \frac{1}{2\pi\sigma^{2}} e^{-(x^{2}+y^{2})/2\sigma^{2}} \tag{3}$$

The complexity of the algorithm is generally important when considering the processing speed of an image processing technique, but this speed is also influenced by the size of the processing image resolution. Face recognition techniques can be performed smoothly at 400,000 pixels, but once an image exceeds 1 million pixels real-time processing cannot be guaranteed. The proposed multiresolution face recognition method adopts the average sizes of the high-resolution and low-resolution images as the standard for the images that

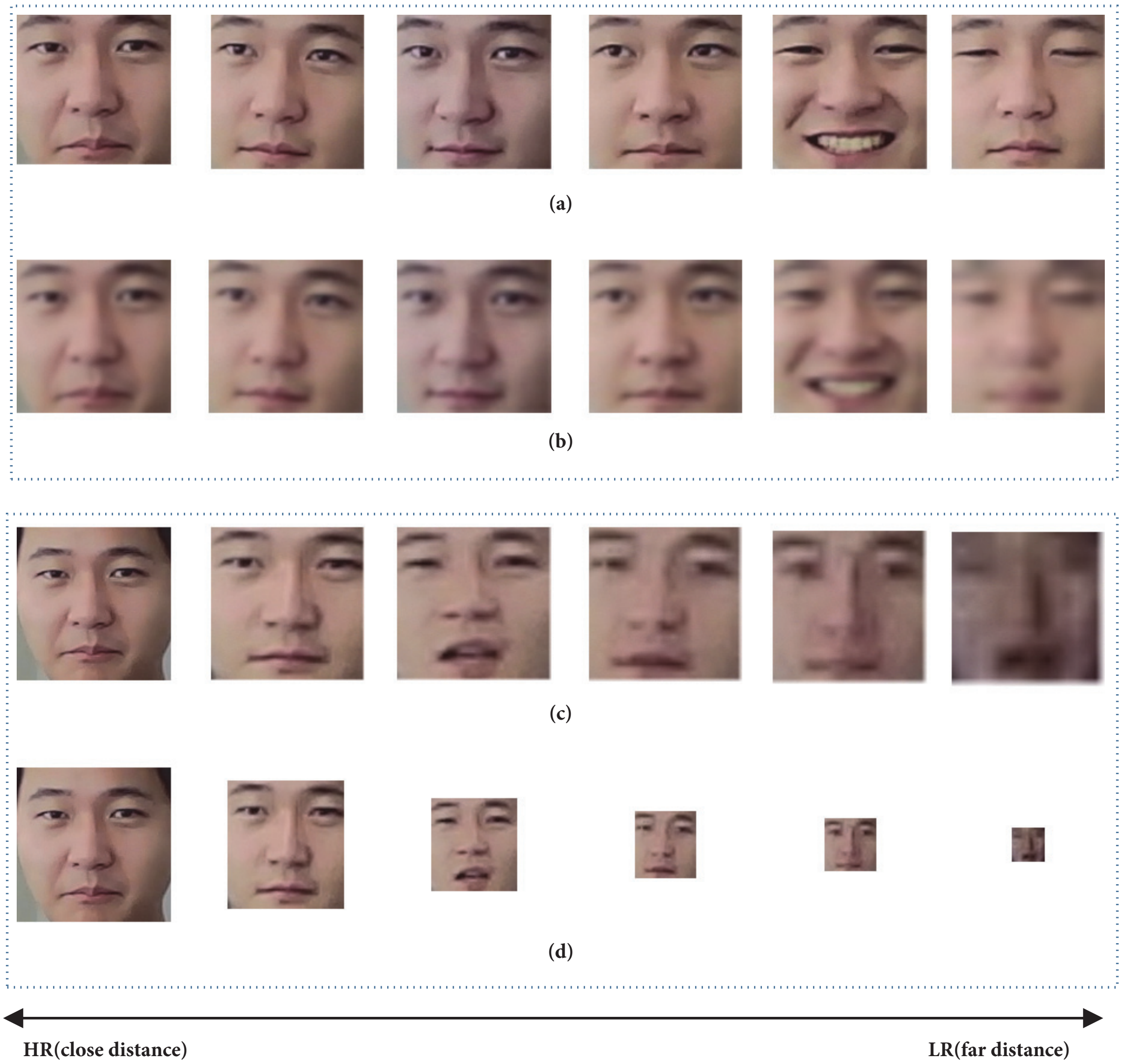


Figure 4: Comparison of real multiresolution images with proposed virtual multiresolution images.

are input for face recognition, instead of using the size of a high-resolution face image as the standard, as is the case in existing techniques. In other words, the sizes of all face images used for training and testing are scaled to the same standard size. The average size of the faces used for face recognition is calculated through the following.

$$RF_{size} = \frac{HR\_width_{max} + LR\_width_{min}}{2} \tag{4}$$

where RF_{size} is the standard size that all face images are normalized to, HR_width_{max} is the width of the highest face image resolution, and LR_width_{min} is the width of the lowest resolution.

3.3. Illumination Normalization. In this paper, histogram interpolation was used to normalize the brightness of faces. A histogram constitutes a method of assessing the distribution of the illumination, which is an important element of data for an image. Histogram configuration is a method that sets the illumination value from 0 to 255 as the index and accumulates the frequency in the corresponding index table according to the illumination value of each pixel that configures the image. Histogram interpolation is a method that alters an images histogram such that it appears evenly across the entire gray scale area. This method does not merely expand the images histogram, but rather changes the histogram distribution based on the characteristics of the original images histogram. The function that changes the images pixel values is defined in (5), where r is the inputted gray scale and s is the gray scale value that is outputted through the conversion function T. In general, the conversion function T is assumed to be a monotone increasing function, and histogram interpolation can also be described in the form of a monotone increasing function.

$$s = T(r) \tag{5}$$

Based on the above, the histograms of the inputted and outputted images can be represented through $P_{\tau}(r)$ and $P_s(s)$, respectively, as probability density functions [14, 15]. The conversion function of histogram interpolation is defined in a form that responds to the accumulated value of $P_\tau(r)$, and is expressed through (6), where τ is a dummy variable for the integration. The function that is defined by integrating over the probability density function is called the cumulative distribution function.

$$s = T(r) = \int_0^r P_r(\tau)\, d\tau \tag{6}$$

3.4. Feature Extraction. PCA has several limitations that are inherent from its nature. Among others, the biggest is that it is ineffective at separating objects, although it is good at summing up data. Discrimination between objects is important, because face recognition aims to distinguish objects. Therefore, we need to know whether changes in a face image occur because the object changes or because of changes in the illumination or face expression. LDA separates different components into groups, and discriminates between changes in face components and changes resulting from other factors [10]. LDA consists of a linear transformation, which maximizes the ratio of the between-class scatter matrix to the within-class scatter matrix and reduces the dimension of a specific vector for the data. The between-class scatter matrix SB and within-class scatter matrix SW can be written as follows:

$$S_B = \sum_{i=1}^{C} (\mu_i - \mu)(\mu_i - \mu)^T \tag{7}$$

$$S_W = \sum_{i=1}^{C} \sum_{j=1}^{K_j} \left(I_j^i - \mu_i\right)\left(I_j^i - \mu_i\right)^T \tag{8}$$

where C is the number of objects, μ_i is the mean image for each object, and μ is the mean image of all images. K_j is the number of the image of object in i sequence. Therefore, W where S_B becomes the maximum and S_W becomes the minimum can be described as below.

If eigenvector and eigenvalue calculated from (7) and (8) are applied to (9), optimal eigenvector can be calculated that shows the level of group's discrimination. This process is repeated when a new image is entered to calculate weight and face recognition is performed by comparing the weight of images in the database and that of a new image. LDA is widely used for research related to face recognition as it clearly discriminates the features of each group.

$$W = \underset{w}{\operatorname{argmax}} \left| \frac{W^T S_B W}{W^T S_w W} \right| \tag{9}$$

3.5. Similarity Measurement. In order to verify the recognition rates for long range face recognition systems, the similarities between feature matrices were compared in order to calculate the recognition rates. For two feature matrices $X = (x_{ij})_{p\times q}$ and $Y = (y_{ij})_{p\times q}$, their similarity can be calculated using the following.

$$d(X, Y) = \sum_{j=0}^{q} \sqrt{\sum_{i=0}^{p} \left(x_{ij} - y_{ij}\right)^2} \tag{10}$$

After converting a random verification image A_t into a feature matrix X_t, the similarity with the feature matrix of all the training data is measured. The feature matrix with the least similarity is determined as shown in (11), and the class inside this feature matrix X_t is determined as the final class of the verification image. The final face recognition is determined by the proportion of verification images T_s that accurately include the class compared to the total number of verification images T, as is shown in (12), where N is the total number of learning images.

$$\min\left(d\left(X_t, X_k\right)\right), \quad k = 1, 2, 3 \ldots, N \tag{11}$$

$$\mathit{recognition} = \frac{T_s}{T} \times 100\% \tag{12}$$

4. Experimental Results

In traditional face recognition experiments, a face DB such as the Yale DB [16], MIT Face DB [17], or FERET DB [18] is normally used. It is found that existing face DBs include factors such as lighting, face twisting, and changes in face expression as external changes, but they do not consider changes in face image resolution according to changes in distance. However, there is a difference between real low-resolution face images that are extracted from long distances and low-resolution face images that have been reduced to the same size using a high-resolution image. In other words, the high-resolution face images from existing DBs are used after being temporarily reduced, but such images are not suitable for analyzing the performance of face recognition from real multiresolutions. The ETRI face DB that was used in this paper is composed of face images from 10 candidates from a u-Robot test bed environment [19]. Table 1 illustrates the composition of the ETRI face DB used in this experiment. Images for each candidate include changes in lighting and distance.

Changes in lighting are achieved by using indoor lighting, and the distance varies from 1 m to 5 m. The size of the original face image was configured to 50×50, 25×25, 16×16, 12×12, and 10×10 using a high-resolution image. In this paper, face recognition was performed using a 1:N search method, instead of a 1:1 authentication method, and the face image that was most similar out of all of those stored in the DB was categorized through the image verification results.

In this experiment, the DB was configured by directly extracting face areas, assuming that all faces would be detected from the input images regardless of the distance. If faces are manually extracted, then face areas can be extracted with higher precision than in an automatic face detection method. In addition, the original image was employed as it was, without considering any twisting or turning in the extracted face images.

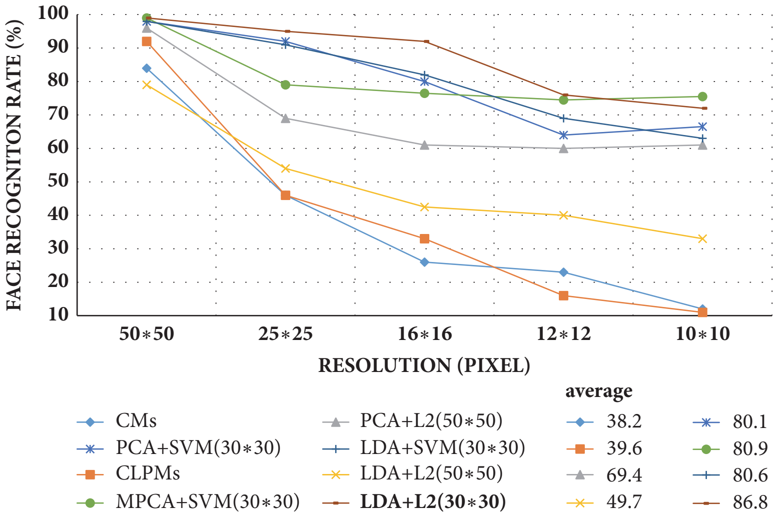


Figure 5: Comparison of recognition accuracies for different single sample face recognition method in ETRI DB.

Table 1: ETRI face DB.

Description	Summary
Environment of Acquisition	(i) 1m 5m distance change (ii) face position change (iii) face expression change
Acquisition Method Resolution	(i) frame division through video (ii) face position change
Distance of Acquisition	(i) 1m~5m(1m intervals)
Total Candidate Count	(i) 10 people
Number of Faces per Person (Average Face Size)	(i) 1m: 30 images(50×50) (ii) 2m: 30 images(25×25) (iii) 3m: 30 images(16×16) (iv) 4m: 30 images(12×12) (v) 5m: 30 images(10×10)

The performance of the proposed method was analyzed by comparing it with PCA, LDA, and MPCA [20], as well as CMs [5] and CLPMs [6], which recognize low-resolution faces by considering structural features. The experiments were all conducted under the same conditions, and only one original face data image was used in training for each candidate. There were 30 verification images for each distance, giving a total of 150 images, and for all candidates there were a total of 1,500 verification images, where learning images were not included in all verification images. Figure 5 presents the face recognition rate change when virtual face images for training. In the experiment results, if a single sample was used then all high-resolution face recognition methods achieved an excellent performance. However, as the resolution decreased there was a significant difference in the performances. In terms of the average recognition rate for all resolutions, the proposed method, which used LDA, achieved the highest performance, at 86.8%. When the size of the standard image was based on the average resolution (30×30) instead of high-resolution (50×50), the average face recognition performance was at an excellent level of 40.7%, even at low-resolutions of 16×16, 12×12, and 10×10. When the face recognition was performed using a single sample in multiresolutions with face images generated for each virtual resolution, LDA performed better than PCA. During the process of extracting features, using the average size of all resolutions performed better for improving the general recognition rate than using the face size based on a high-resolution as the standard. Moreover, on account of issues occurring when the number of pixels in an image is higher than the number of images, LDA achieved a better performance than existing techniques such as CLPMs and MPCA.

5. Conclusions

In camera-based face recognition, various resolutions can exist, in terms of face images that are acquired in conditions from close range high-resolution images to long range low-resolution images. If high-resolution images are converted into low-resolution images for face recognition, then the face recognition performance may suffer on account of a loss of data and a difference between the resolution of the face image that was used for training and that of the verification image. This paper has proposed a method that recognizes faces by generating face images for each virtual resolution in order to resolve the issue of reduced recognition rates resulting from changes in resolution caused by changes in the distance between the camera and the subject. The proposed method uses one high-resolution image to automatically generate face images for each resolution to use for training. The face image size is normalized to the average size of all faces and used for face recognition. The proposed method achieved the highest performance of the tested methods, with an average resolution of 86.8%, when LDA and the Euclidean distance were used. Moreover, when the standard image size was set to 30×30, which is the average image size, the performance improved by an average resolution of 37.1% compared with when the standard image size was set to 50×50, under the same experimental conditions.

Data Availability

The ETRI DB data used to support the findings of this study have not been made available due to Electronics and Telecommunications Research Institute under license and so cannot be open to the public.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (no. 2018R1A2B6001984) and supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. 2017R1A6A1A03015496).

References

[1] H.-M. Moon, C. H. Seo, and S. B. Pan, “A low-cost media quality enhancement resolution up-conversion for mobile cloud,” *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3098–3111, 2017.

[2] W. W. Zou and P. C. Yuen, “Very low resolution face recognition problem,” *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 327–340, 2012.

[3] Z. Wang, Z. Miao, Q. M. Jonathan Wu, Y. Wan, and Z. Tang, “Low-resolution face recognition: a review,” *The Visual Computer*, vol. 30, no. 4, pp. 359–386, 2014.

[4] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[5] S. Baker and T. Kanade, “Hallucinating faces,” in *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000*, pp. 83–88, Grenoble, France, March 2000.

[6] J. Liu, J. Qiao, X. Wang, and Y. Li, “Face hallucination based on independent component analysis,” in *Proceedings of the 2008 IEEE International Symposium on Circuits and Systems, ISCAS 2008*, pp. 3242–3245, Seattle, Wash, USA, May 2008.

[7] B. Li, H. Chang, S. Shan, and X. Chen, “Coupled metric learning for face recognition with degraded images,” in *ACML 2009: Advances in Machine Learning*, vol. 5828 of *Lecture Notes in Computer Science*, pp. 220–233, 2009.

[8] B. Li, H. Chang, S. Shan, and X. Chen, “Low-Resolution Face Recognition via Coupled Locality Preserving Mappings,” *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 20–23, 2010.

[9] P. Salvagnini, L. Bazzani, M. Cristani, and V. Murino, “Person re-identification with a PTZ camera: An introductory study,” in *Proceedings of the 2013 20th IEEE International Conference on Image Processing, ICIP 2013*, pp. 3552–3556, Melbourne, Australia, September 2013.

[10] U. Park, H.-C. Choi, A. K. Jain, and S.-W. Lee, “Face tracking and recognition at a distance: a coaxial and concentric PTZ camera system,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 10, pp. 1665–1677, 2013.

[11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 3rd edition, 2008.

[13] F. Crete, T. Dolmiere, P. Ladret, and M. Nicolas, “The blur effect: perception and estimation with a new no-reference perceptual blur metric,” in *Proceedings of the Human Vision and Electronic Imaging XII*, vol. 6492 of *Proceedings of SPIE*, San Jose, Calif, USA, February 2007.

[14] E. Parzen, “On estimation of a probability density function and mode,” *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[15] V. A. Epanechnikov, “Non-parametric estimation of a multivariate probability density,” *Theory of Probability & Its Applications*, vol. 14, pp. 153–158, 1969.

[16] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: illumination cone models for face recognition under variable lighting and pose,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[17] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz, “Component-based face recognition with 3D morphable models,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2004*, Washington, Wash, USA, July 2004.

[18] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[19] D. Kim, J. Lee, H.-S. Yoon, and E.-Y. Cha, “A non-cooperative user authentication system in robot environments,” *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 804–811, 2007.

[20] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “MPCA: multilinear principal component analysis of tensor objects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 1, pp. 18–39, 2008.