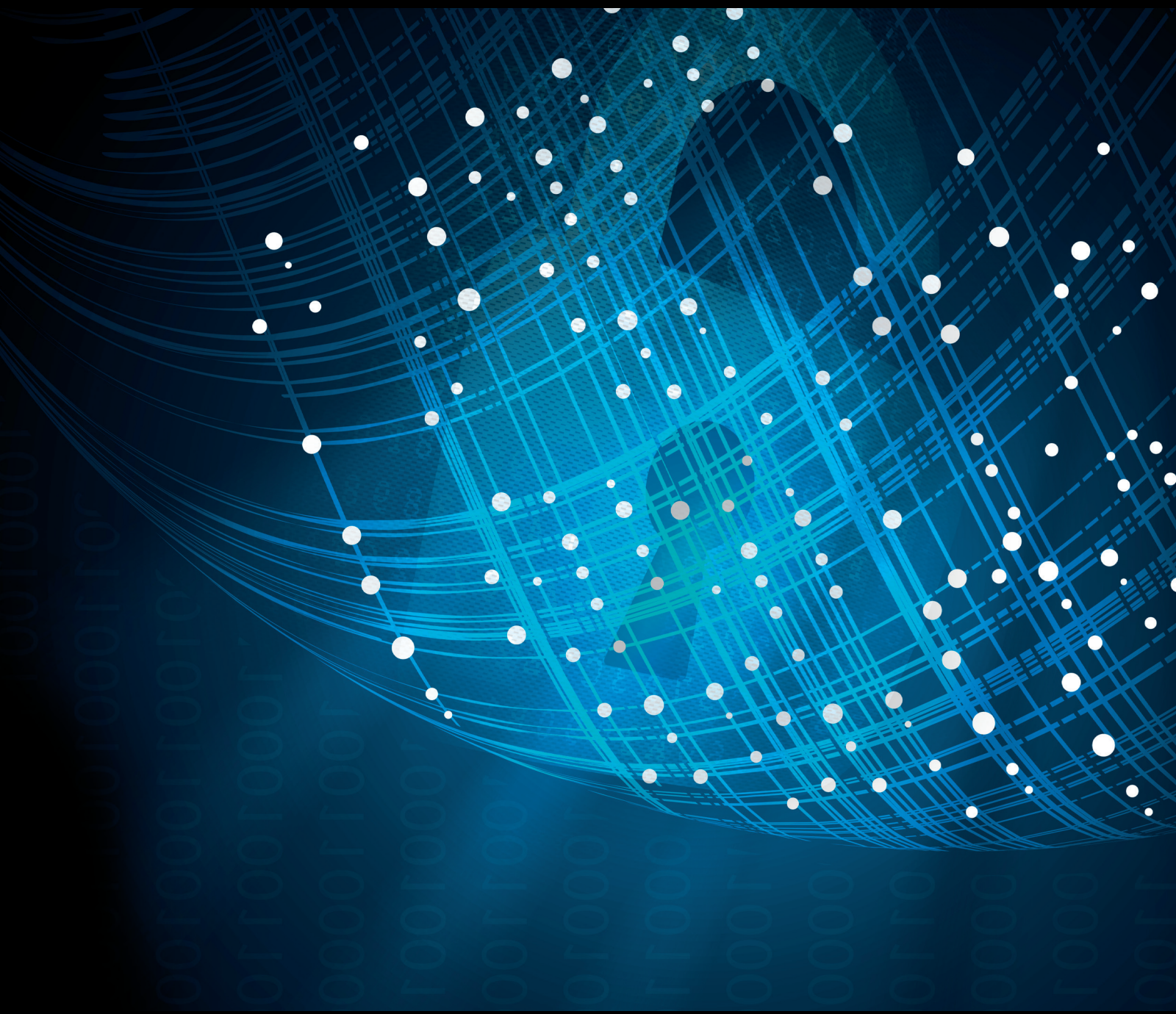
Security and Communication Networks

Security and Privacy in Internet of Medical Things (IoMT)

Lead Guest Editor: Geethapriya Thamilarasu
Guest Editors: Kewei Sha, Kuan Zhang, and Wenjia Li



Security and Privacy in Internet of Medical Things (IoMT)

Security and Communication Networks

Security and Privacy in Internet of Medical Things (IoMT)

Lead Guest Editor: Geethapriya Thamilarasu
Guest Editors: Kewei Sha, Kuan Zhang, and Wenjia Li

Contents

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8014958, 11 pages
https://doi.org/10.1155/2020/8014958

WILEY | Hindawi

Research Article

Abnormal User Detection Based on the Correlation Probabilistic Model

Xiaohui Yang and Ying Sun

School of Cyber Security and Computer, Hebei University, Baoding, China

Correspondence should be addressed to Xiaohui Yang; yxh@hbu.edu.cn

Received 25 October 2019; Revised 1 February 2020; Accepted 24 February 2020; Published 13 June 2020

Guest Editor: Wenjia Li

As an important part of the new generation of information technology, the Internet of Things (IoT), which is developing rapidly, requires high user security. However, malicious nodes located in an IoT network can influence user security. Abnormal user detection and correlation probability analysis are fundamental and challenging problems. In this paper, the probabilistic model of the correlation between abnormal users (PMCAU) is proposed. First, the concept of user behavior correlation degree is proposed, which is defined as two parts: user attribute similarity degree and behavior interaction degree; the attribute similarity measurement algorithm and behavior correlation measurement algorithm are constructed, respectively, and the spontaneous and interactive behaviors of users were analyzed to determine the abnormal correlated users. Second, first-order logic grammar is used to express the before and after connection of user behavior and to deduce the probabilistic of occurrence of the correlation of behavior and determine the abnormal user groups. Experimental results show that, compared with the traditional anomaly detection algorithm and Markov logic network, this model can identify the users correlated with anomalies, make probabilistic inferences on the possible associations, and identify the potential abnormal user groups, thus achieving higher accuracy and predictability in the IoT.

1. Introduction

Internet of Things (IoT) is the latest evolution of the Internet, including a great deal of connected physical devices and applications [1]. IoT allows object collection, data exchange, etc. [2], which can perform medical data management, medical information monitoring, and user information analysis. IoT is an open network, and there are a large number of malicious nodes in the network. These malicious nodes may tamper with the correct data and pass them to other nodes. The normal nodes will use the wrong data for information dissemination due to the lack of ability to verify the correctness of the messages received, resulting in the dissemination of false information on medical, social, and other networks. Network individuals form a "relationship structure" through various connection relationships, including virtual communities composed of various complex relationship associations; [3] based on the relationship structure, a large number of network individuals aggregate and influence each other around an event to form a "network group" with common behavior features; based on the relationship structure and network group, all kinds of information can be quickly released and spread to form social media, feedback and act on the real society, so that the interaction the real society has a great impact on the real world.

To detect abnormal user groups in the IoT, the probabilistic model of correlation between abnormal users (PMCAU) is proposed. PMCAU studies the daily behavior of malicious users, considers the behavior information and interaction information of malicious users, constructs the attribute similarity measurement algorithm and behavior correlation measurement algorithm, calculates various information of malicious users respectively, and finds abnormal correlated users. At the same time, probabilistic soft logic is used to predict the possibility of abnormal

correlation in the future, to identify the potential abnormal user groups in the IoT.

The contributions of this paper are listed as follows:

(1) Construct attribute similarity measurement algorithm, read user attribute data, and calculate user attribute similarity, including geographic location similarity, user follower similarity, and personal information similarity.

(2) Construct behavioral interaction measurement algorithm, consider user interaction behavior information, calculate the degree of interaction between user interaction behaviors, and reflect the features of user interaction behavior.

(3) Propose the concept of behavioral correlation degree as an important distinguishing indicator between abnormal correlated users and nonabnormal correlated users. At the same time, the correlation threshold is defined, correlation threshold judgment based on behavioral correlation degree to identify abnormal correlated users.

(4) Use probabilistic soft logic to express the causes and consequences of abnormal behavior correlation in abnormal user groups and combine multiple factors such as geographic location, behavior information, interest and preference, and fans' attention information to propose a set of reasoning rules more suitable for predicting the probabilistic of correlation among abnormal correlated users and to identify the potential abnormal user groups.

(5) Analyze the performance of PMCAU in the real data set and compared it with other algorithm models. PMCAU has better performance in improving the accuracy, stability, and probabilistic reasoning of identifying abnormal correlated users.

The rest of this paper is organized as follows: In Section 2, we analyze and explain existing related work and theoretical basis. The model is described in Section 3. In Section 4, we introduce the proposed model PMCAU in detail. In Section 5, we present the experimental results. Finally, we conclude our work in Section 6.

2. Related Work

Users are the main part of IoT. Accurate analysis of their behavioral correlation and anomaly detection and inference prediction are necessary to maintain network security. For abnormal detection in the IoT, scholars mostly adopt analysis methods based on network traffic behavior [4, 5] and analysis methods based on host behavior anomalies, such as network behavior modeling through host audit command [6]. But this class of methods only considers the behavior of a single user, without considering the influence of the user's own attributes, and hence it has lower detection accuracy. Recently, research results have been obtained by analyzing users' daily behaviors to predict their later behaviors and detect abnormal behaviors of users. Cao et al. [7] statistically analyzed various factors that may affect user behaviors and used logistic regression, Bayesian network, and other methods to predict user forwarding behaviors by virtue of user attributes, social relations, and other characteristics. Xu et al. [8] used a hybrid implicit topic model to predict user forwarding behavior based on the features of the user's forwarding behavior being affected by factors such as breaking news and users' own interests. The above algorithms do not take into account the multiple behaviors of users and the correlation between behaviors, so it cannot comprehensively judge and deal with abnormal behaviors of users' attribute information and behavioral information, and its reliability and accuracy are often difficult to be guaranteed. The PageRank-based account anomaly detection algorithm [8] builds a social relation matrix based on the user relationship and ranks the account to detect malicious users through the iterative calculation of PageRank value. Because this method does not consider the user's own attribute characteristics, the ranking result of the user is affected by the time delay, so the accuracy rate is low in the heterogeneous network with complex structure and uneven scale. The Markov logic network is a statistical relation learning model combining Markov network and first-order logic [9, 10]. Although it has certain reasoning abilities for correlation probability of microblog users, its accuracy is low because it uses Boolean value {0, 1} instead of continuous value [0, 1] to determine whether there is an abnormal correlation. Sun et al. [11] proposed a joint fraud detection method based on abnormal user groups and mining abnormal user groups through the similarity adjacency graph. Yang et al. [12] proposed a step-by-step detection method to find the anomaly level, constructed a scoring matrix, and used the fast maximum margin matrix decomposition to perform level prediction to capture anomalies. These algorithms cannot accurately reflect the behavior features of users, cannot adapt to the features of microblog data information, and cannot make probabilistic reasoning on the occurrence of behavioral correlation, so the accuracy of detecting abnormal user groups is low.

In summary, existing anomaly group detection methods have two important defects. First, the methods based on the network popularity or behavior prediction model only consider the behavior information sent by the user. It is impossible to comprehensively judge and process abnormal behaviors of user attribute information and behavior information, and its reliability and accuracy are often difficult to guarantee. Second, the user's multiple behaviors and the relationship between behaviors are not taken into consideration, and anomaly detection and probabilistic reasoning of the group cannot be taken into consideration, and the accuracy of detecting potential abnormal user groups is low.

When objects connected to the Internet of Things continue to generate information and report to Internet users, a noteworthy development is that they will also join traditional social networks and interact with "people" in social networks. Social networks are not just person-to-person social, but person-to-person, person-to-thing, and thing-to-thing. Therefore, abnormal user groups in social networks will inevitably pose a threat to the security of the

Internet of Things. At this time, the Internet of Things to hardware also has social attributes. Therefore, to maintain the security of the Internet of things and detect abnormal user groups in the network, in response to the above problems, the probabilistic model of correlation between abnormal users (PMCAU) is proposed by taking the social platform of microblog with a large user volume as an example. PMCAU defines the concept of behavioral correlation degree and constructs an attribute similarity measurement algorithm and a behavioral interaction measurement algorithm to identify abnormal correlated users in microblog and to perform probabilistic reasoning in abnormal correlated users to complete the detection of abnormal user groups.

3. Model Description

To accurately identify abnormal users and find potential abnormal user groups before the occurrence of threat events, a probabilistic model of correlation between abnormal users (PMCAU) is proposed. The key of PMCAU is to analyze the behavior correlation between malicious users, calculate the behavior correlation degree of malicious users, determine the abnormal correlated user, and complete the probabilistic reasoning for users with different behavior correlation degree. $UM = \{um_i\}(i = 1, \ldots, n)$ represents a collection of malicious users, namely the zombie users, spam users, compromised users, etc. And $acu \in U$ represents an abnormal correlated user identified in the microblog. Here, the abnormal correlated user refers to a special group of junk users who publish specific information for specific content in the network, who are organized to publish, reply to, and forward blog posts or refer to others, to quickly spread bad and instructive error information. Not only let normal users cannot see the truth of the event but also will cause misleading to normal users, with adverse social consequences. The model framework is shown in Figure 1:

(1) *Data Layer*. Read the original data and preprocess the data. The user vector is constructed, and the valid user attribute information and user blog text information in the original data are selected.

(2) *Feature Layer*. Analyze user attribute information and behavior information based on user features. Extracting attribute features of the user's daily behavior based on the geographical distribution of the user's self-issued location, the number of followers, and personal data; the interaction behavior features of the user are extracted based on the occurrence object of the interaction behavior between users, the frequency of interaction, and the like.

(3) *Detection Layer*. Two algorithms are proposed to detect abnormal correlated users. The attribute similarity measurement algorithm calculates the attribute similarity between users' spontaneous behaviors, including three aspects: geographical similarity, follower's information, and personal information similarity. The behavioral interaction measurement algorithm constructs a user interaction degree formula, which indicates the strength of the user correlation process and is used to calculate the degree of correlation between user interaction behaviors.

(4) *Reasoning Layer*. The first-order logic grammar is used to express the causal connection before and after the abnormal behavior occurs and to reason the probabilistic of the connection and determine the abnormal user groups.

PMCAU detects potential malicious threats by determining abnormal correlated users *acu* and probabilistic reasoning of correlation degree between abnormal behaviors. Before the occurrence of a threat event, it can quickly mark abnormal user groups and find possible attack behaviors in advance to ensure the healthy and smooth operation of the network.

The abnormal behavior of users in the network seriously damages the network security. Due to the increasing diversification and concealment of user behaviors in microblog, PMCAU detects microblog users from the aspects of behavioral correlation analysis and abnormal correlated probabilistic reasoning. Behavioral correlation analysis is used to calculate the user's attribute similarity and behavioral interaction degree and to determine the abnormal correlated users *acu*; Abnormal correlation probabilistic reasoning is used to calculate the probability of future correlation between abnormal correlated users, and the probabilistic of occurrence of abnormal correlation is determined by the probabilistic, to discover potential groups threats.

4. Behavioral Correlation Analysis

Security of microblog users depends not only on their security but also is closely related to the behavior between users. By defining and calculating the behavioral correlation degree, users with a behavioral correlation greater than the correlation threshold φ are determined as abnormal correlated users *acu*.

Attribute similarity (AS) measures the similarity of three attributes, such as the user's geographical location, the number of user followers, and the integrity of personal information. Behavior interaction (BI) represents the interaction frequency of users, which are reflected by the interaction behavior between users. Behavior correlation degree (BC) represents the correlation strength of malicious user behaviors, which are obtained by calculating malicious user attribute similarity AS and behavior interaction BI. The calculation formula is shown in the following equation:

$$\mathrm{BC}\left(um_i, um_j\right) = \theta \mathrm{AS}\left(um_i, um_j\right) + (1 - \theta)\mathrm{BI}\left(um_i, um_j\right), \tag{1}$$

where θ represents the harmonic coefficient and the specific value is determined by the experiment. BC (u_i, u_j) represents the degree of correlation of users' behaviors.

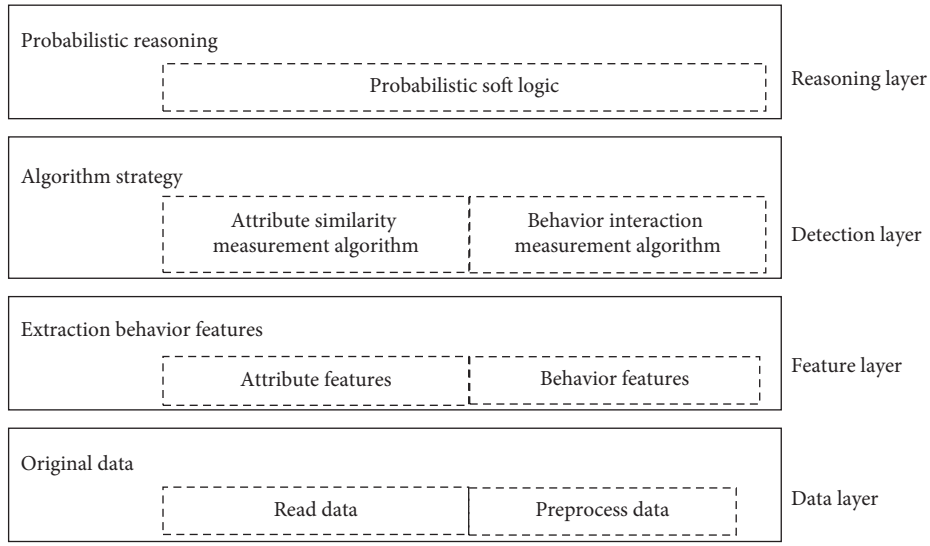


Figure 1: PMCAU framework.

4.1. Attribute Similarity. For malicious users in microblog, it is very important to study their geographical location, number of followers, and personal information. If malicious users living in the same area have a large number of overlapping users and similar personal information, it indicates that there may be some links between these malicious users. Jaccard coefficient is used to calculate the similarity of user attributes, and the relationship between users is obtained. The user attribute features are shown in Table 1.

The Jaccard coefficient is used to calculate the similarity between samples of a symbol metric or a Boolean metric and to compare similarities and differences between finite sample sets. Given two sets A and B, the similarity is measured by the ratio of the intersection of the two sets and the union. The larger the Jaccard coefficient, the higher the sample similarity. Since the Jaccard coefficient is suitable for calculating the similarity of discrete sets, the values of each element in its scoring matrix are expressed as 1 (with) and 0 (without) to determine whether there are common features between samples, which has a good calculation effect. The formula is shown in the following equation:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{2}$$

Users in microblog are affected by their attributes, such as users' geographical location, user followers' coincidence degree, and personal information, and their interactive relationship implies a certain real behavioral connection. The Jaccard coefficient was used to calculate the similarity of its three attributes of geographical location A_g, number of followers A_f, and personal information A_c and analyze the correlation strength of behaviors among malicious users. AS (um_i, um_j) is the attribute similarity of malicious users um_i and um_j, and the higher its value is, the higher the attribute similarity is; the formula is as shown in the following equations:

$$J_1\left(um_i\left(A_g\right), um_j\left(A_g\right)\right) = \frac{\left|um_i\left(A_g\right) \cap um_j\left(A_g\right)\right|}{\left|um_i\left(A_g\right) \cup um_j\left(A_g\right)\right|}, \tag{3}$$

$$J_2\left(um_i\left(A_f\right), um_j\left(A_f\right)\right) = \frac{\left|um_i\left(A_f\right) \cap um_j\left(A_f\right)\right|}{\left|um_i\left(A_f\right) \cup um_j\left(A_f\right)\right|}, \tag{4}$$

$$J_3\left(um_i\left(A_c\right), um_j\left(A_c\right)\right) = \frac{\left|um_i\left(A_c\right) \cap um_j\left(A_c\right)\right|}{\left|um_i\left(A_c\right) \cup um_j\left(A_c\right)\right|}, \tag{5}$$

$$\begin{aligned} \mathrm{AS}\left(um_i, um_j\right) = 1 - \Big[& J_1\left(um_i\left(A_g\right), um_j\left(A_g\right)\right) \\ & \times J_2\left(um_i\left(A_f\right), um_j\left(A_f\right)\right) \\ & \times J_3\left(um_i\left(A_f\right), um_j\left(A_f\right)\right)\Big]. \end{aligned} \tag{6}$$

4.2. Behavior Interaction. The interaction of microblog users mainly includes 7 behaviors, such as following, forwarding, commenting, thumbs up, collecting, mentioning @, and private letter. When calculating the behavioral interaction degree of malicious users, their login time and online time is different. Therefore, a time scale should be considered the maximum time malicious user interactions with other users of the interval rather than simply calculate the interaction between malicious users in a specified period.

Maximum time interval (MIT) represents the maximum time interval between the malicious user um_i and other users, and this is taken as the time label to measure the interaction frequency of malicious users.

In this regard, mutual behaviors of um_i and um_j in their respective MTI should be taken into consideration together to comprehensively analyze the behavioral interaction degree of um_i and um_j. Calculate the ratio between interaction frequency of um_i against um_j in MTI (um_i), and the interaction frequency of um_i against all malicious users in MTI

Table 1: User attribute features.

Feature symbol	Feature category	Feature name
A_g	User attribute	Geographical location
A_f	User attribute	Number of followers
A_c	User attribute	Personal information

(um_i, um_j). The ratio of um_i to um_j in MTI (um_j) to um_j to all malicious users in MTI (um_i, um_j) is calculated. And the two ratios are added together. The formula is shown in the following equation:

$$\mathrm{BI}\left(um_i, um_j\right) = \sum_{b_i} b_i\left(um_i, um_j\right), \tag{7}$$

where b_i represents 7 interactive behaviors among malicious users: follow, forward, comment, thumbs up, favorite, mention @, and private letter. BI (um_i, um_j) calculates the degree of interaction according to user interaction behavior and obtains the degree of association of behaviors sent by malicious users, as shown in the following equations:

$$\mathrm{inter}\left(um_i, um_j\right) = \frac{\mathrm{NBI}\left(um_i, um_j\right)/\mathrm{MTI}\left(um_i, um_j\right)}{\mathrm{NBI}\left(um_i\right)/\mathrm{MTI}\left(um_i\right)} + \frac{\mathrm{NBI}\left(um_i, um_j\right)/\mathrm{MTI}\left(um_i, um_j\right)}{\mathrm{NBI}\left(um_j\right)/\mathrm{MTI}\left(um_j\right)}, \tag{8}$$

$$b_i\left(um_i, um_j\right) = \frac{\mathrm{inter}\left(um_i, um_j\right)}{\mathrm{inter}_{\max}\left(um_i, um_j\right)}, \tag{9}$$

where NBI (um_i, um_j) represents the number of times the malicious user um_i interacts with um_j; NBI (um_i) and NBI (um_j) represent the total number of times that user um_i and um_j interact bi with all users. MTI (um_i, um_j) represents the maximum time interval in which a malicious user um_i interacts with um_j, MTI (um_i) and MTI (um_j), respectively, represent the maximum time interval for malicious user um_i and um_i to interact b_i with all users.

5. Abnormal Correlation Probabilistic Reasoning

Correlation of abnormal users involve a series of tedious processes such as behavior analysis, prediction, and judgment, and users have many uncertain factors such as incomplete login time, space, and behavior records. Using probabilistic soft logic (PSL) to express the cause and effect of behaviors, combining with many factors such as geographical location, behavioral information, interest preference, and user follower information, a set of reasoning rules which are more suitable to predict the possibility of correlation between abnormal behaviors are proposed and encoded in the PSL framework. PSL has efficient reasoning capabilities that allow the use of first-order logic to specify probabilistic models, which are an expression of if-then rules and support user groups.

5.1. PSL Grammar. Rule composition in PSL is shown in the following equation:

$$P_1(\alpha, \gamma) \wedge P_2(\gamma, \delta) \gg P_2(\alpha, \delta)\text{: weight}. \tag{10}$$

P_1 and P_2 are the predicates that define the relationship between random variables α, γ, and δ; "weight" represents the weight, representing the importance of each rule in reasoning; and ">>" represents the pointing relation of rule body to rule header in PSL [13].

For example, Relationship ($B(acu_i, acu_j)$, Fut) represents whether there is a relationship between $B(acu_i, acu_j)$ in Fut. Correlation ($B(acu_i, acu_j)$, Hist) represents that there will be a correlation between $B(acu_i, acu_j)$ in Hist. Based on the combination of the predicates Relationship and Correlation, the probability of a correlation between $B(acu_i, aca_i)$ in Fut can be expressed, where $B(acu_i, acu_j)$ represents the behavior of the exception correlated user acu_i to acu_j; Fut represents the future period; and Hist represents the historical period.

5.2. Rule Construction. In general, the probabilistic reasoning model uses explicit correlative reasoning and uses Boolean values {0, 1} to determine whether the two are related; "1" for correlation and "0" for no correlation. For example, if it is known that abnormal correlated users acu_i and acu_j have a correlation relationship in the historical period, then the relationship between acu_i and acu_j is still judged as "1" in the future period, that is, there is a correlation relationship in the future period. However, in reality, due to the differences in information such as microblog users' location, active time, interactive behavior, mutual attention, and mutual attention, the accuracy of reasoning is low if only relying on simple explicit correlation rules. Therefore, it is necessary to optimize the inference rules to carry out implicit correlation reasoning. This kind of implicit correlation inference rules make the expression of inference results more in line with the actual situation and have higher accuracy. According to each rule obtained by weight learning, confidence is allocated as weight built in PSL. The rules are as follows.

The more the same locations are located, the greater the likelihood of correlation will be as follows:

$$\mathrm{position}\left(B\left(acu_i, acu_j\right), Po\right) \wedge \mathrm{position\ judge}(Po) \Longrightarrow \mathrm{strong\ correlation}\left(B\left(acu_i, acu_j\right)\right). \tag{11}$$

The higher the coincidence degree of online active time, the higher the possibility of correlation as follows:

$$\mathrm{time\ coincidence}\left(B\left(acu_i, acu_j\right), \mathrm{TC}\right) \wedge \mathrm{time\ coincidenc\ judge}(\mathrm{TC}) \Longrightarrow \mathrm{strong\ correlation}\left(B\left(acu_i, acu_j\right)\right). \tag{12}$$

The higher the frequency of user interaction, the greater the likelihood of correlation:

$$\mathrm{behavior\ interaction}\left(B\left(acu_i, acu_j\right), \mathrm{BI}\right) \wedge \mathrm{behavior\ interaction\ judge}(\mathrm{BI}) \Longrightarrow \mathrm{strong\ correlation}\left(B\left(acu_i, acu_j\right)\right). \tag{13}$$

The more users with the mutual concern, the greater the possibility of correlation:

$$\text{mutual concern}\left(B\left(acu_i, acu_j\right), \text{SF}\right) \wedge \text{mutual concernjudge}\left(\text{SF}\right) \Longrightarrow \text{strong correlation}\left(B\left(acu_i, acu_j\right)\right). \tag{14}$$

Focus on each other can create correlation:

$$\text{focus on each other}\left(B\left(acu_i, acu_j\right), \text{MA}\right) \wedge \text{focus on each other}\left(\text{MA}\right) \Longrightarrow \text{strong correlation}\left(B\left(acu_i, acu_j\right)\right). \tag{15}$$

5.3. Weight Learning. PSL provides maximum probabilistic inference [13] to infer the most likely probabilistic of atoms in logical rules from existing data. Since the continuous value between [0, 1] is used for the probabilistic, MPE reasoning is transformed into a convex optimization process to find the optimal solution.

For the rules learned, the weight of each rule is assigned according to the confidence of each rule. For example, when the correlation between abnormal behaviors is inferred, there are three rules with confidence levels of 0.7, 0.8, and 1.0. When converting them into PSL rules, the confidence can be multiplied by multiple times as its weight. However, for manually defined rules, weight learning is required. In weight learning, the maximum likelihood estimation method [14] is used, and the gradient function is used to perform weight estimation. The formula is as shown in the following equation:

$$\frac{\partial}{\partial \lambda_w} \log_a f\left(Q\right) = -\sum_{r \in R_w} \left(D_r\left(Q\right)\right)^P + E\left[\sum_{r \in R_w} D_r\left(Q\right)\right], \tag{16}$$

where R_w represents all logical rules with weight λw being initialized, $E[\sum_{r \in R_w} D_r(Q)]$ is replaced by a $\sum_{r \in R_w} D_r(Q^*)$ approximation, and Q^* is the most likely correct interpretation of the atom.

5.4. Probabilistic Reasoning. PSL is different from other kinds of probabilistic models and is the feature of closed atoms using soft constraints, namely closed atomic probabilistic values is a continuum of values between [0, 1], rather than Boolean value {0, 1}, the record for $Q(r)$, usually adopt Lukasiewicz logic [13] the conjunction ($\wedge$), disjunction ($\vee$), and negative ($\neg$) as a logical connective to calculate $Q(r)$. The formulas are as shown in the following equations:

$$Q\left(l_1 \wedge l_2\right) = \max\left\{Q\left(l_1\right) + Q\left(l_2\right) - 1, 0\right\}, \tag{17}$$

$$Q\left(l_1 \vee l_2\right) = \min\left\{Q\left(l_1\right) + Q\left(l_2\right), 1\right\}, \tag{18}$$

$$Q\left(l_2\right) = 1 - Q\left(l_1\right). \tag{19}$$

A rule r in PSL can be described as $r\text{body} \longrightarrow r\text{head}$. This rule is satisfied when $Q(r\text{body}) \leq Q(r\text{head})$ or $Q(r) = 1$. Otherwise, the degree of satisfaction of the logic rule is measured by calculating the distance satisfaction $D(r)$, as in the following equation:

$$D\left(r\right) = \max\left\{0, Q\left(r\text{body}\right) - Q\left(r\text{head}\right)\right\}. \tag{20}$$

PSL defines the probabilistic value of the probabilistic distribution for all closed atoms, as in equation (21), and defines the probabilistic value of the closed atom as the distance satisfaction between the highest probabilistic value and the lowest probabilistic value, that is, the probabilistic satisfies all logic rules:

$$p\left(Q\right) = \frac{1}{\text{Z}} \exp\left\{-\sum_{r \in R} \lambda_r D\left(r\right)\right\}, \tag{21}$$

$$\text{Z} = \int_Q \exp\left\{-\sum_{r \in R} \lambda_r D\left(r\right)\right\}, \tag{22}$$

where Z is the normalization constant, λ_r is the weight of the rule r, and R is the set of all rules.

The PSL model input is the abnormal behavior data of malicious users, the input data set is used to initialize the logic rules in the PSL model, and the weight learning is performed. Then, distance satisfaction is defined in the PSL model, and the probabilistic of meeting the logic rules initialized every day is calculated. Finally, MPE reasoning mechanism is applied to calculate the probabilistic correlation between abnormal behaviors.

6. Experiments

6.1. Experimental Environment and Data. The environment used in the experiment was Intel(R) Core(TM) i5-7300HQ CPU @2.50 GHz, 8 GB of memory, the operating system is Windows 10, and model code is based on c++ implementation.

The data set published in literature [15] was used to verify the feasibility of the model. The data set contains 1,787,443 microblog user data, and each of the user data includes basic information of the users (such as user ID, gender, number of followers, number of fans, etc.), 1000 microblogs newly released by each user, and user interaction behavior data. Among them, there are nearly 4 billion relationships of mutual concern among users, and each user has an average of 200 followers. Due to a large amount of data in the data set, 10 groups are randomly selected from the data set, each group has 10,000 pieces of user data, and each piece of user data includes the basic information of the user, the newly published blog content, and user interaction behavior data, which is recorded as "Data1," "Data2," "Data3," "Data4," "Data5," "Data6," "Data7," "Data8," "Data9," and "Data10."

6.2. Evaluation Index. To solve the data imbalance problem, confusion matrix analysis experimental results are established [16]. In the matrix, TP stands for the number of users that are originally abnormal correlated users and are judged to be abnormal correlated users during detection; FN stands for the number of users that are originally abnormal

correlated users but are judged to be nonabnormal correlated users during detection; FP stands for the number of users that are originally nonabnormal correlated users but are judged to be abnormal correlated users during detection; and TN stands for the number of users that are originally nonabnormal correlated users and are judged to be nonabnormal correlated users during detection, as shown in Table 2.

To evaluate the performance of PMCAU, three evaluation indexes, namely, precision rate (Pre), recall rate (Rec), and harmonic mean value $F1$_score, were selected. Among them, the precision rate and recall rate were used to evaluate the accuracy of the experiment, and the harmonic mean value $F1$_score was used to evaluate the comprehensive performance of the experiment. Pre refers to the proportion of the number of correctly identified abnormal correlated users of all detected abnormal correlated users. Rec refers to the proportion of correctly identified abnormal correlated users in the total number of truly abnormal correlated users. $F1$_score is the harmonic mean value of precision rate and recall rate, and the equation is as follows:

$$\begin{aligned} \text{Pre} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Rec} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F1\_\text{score} &= 2 \cdot \frac{\text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}}. \end{aligned} \tag{23}$$

6.3. Parameter Settings. Parameters involved in the experiment include the linear coefficient θ (used to reconcile the user's attribute similarity and behavioral interaction) and the correlation threshold φ; their values are determined experimentally, and its value was determined by referring to the performance of the model evaluated utilizing harmonic mean $F1$_score.

For malicious users, the definition of their abnormal correlated users is the key to detect abnormal groups in microblog. Linear coefficient θ and correlation threshold φ are directly related to the detection of abnormal correlated users. According to formula (9), different linear coefficients θ correspond to different correlation degree thresholds. $F1$_score values of correlation threshold φ under different values of linear coefficient θ were calculated, respectively, as shown in Figure 2. According to experience, the degree of behavioral interaction between malicious users has a greater impact than the degree of attribute similarity. Therefore, the value range of the linear coefficient θ is set as [0.15, 0.55] and the value range of the correlation threshold is set as [0.1, 0.9].

Figure 2 represents different $F1$_score with the value range of linear coefficient θ set as [0.15, 0.55] and the correlation threshold φ with a value range of [0.1, 0.9]; when the linear coefficient $\theta = 0.35$ and the correlation threshold $\varphi = 0.7$, $F1$_score is the largest. Therefore, it can be considered that in the calculation of behavioral correlation degree, when the linear coefficient θ is set at 0.35, the correlation threshold φ for defining abnormal behaviors is set at 0.7. That is, when users with a behavior correlation degree greater than 0.7 in malicious users are defined as abnormal correlated users, PMCAU has better detection performance.

The distribution of user behavior correlation degree and abnormal correlation probabilistic are shown in Figures 3 and 4. After the parameter setting, the behavioral correlation degree of abnormal correlated users is distributed in [0.7, 1.0] and that of nonabnormally correlated users is distributed in [0.2, 0.6]. Meanwhile, when reasoning the abnormal correlated probabilistic of users, the correlation probabilistic of abnormal correlated users is [0.8, 1.0] and that of nonabnormal correlated users is [0.5, 0.7]. The above indicates that when the linear coefficient θ is set to 0.35 and the correlation threshold φ for defining abnormal behaviors is set to 0.7, PMCAU can better classify abnormal correlated users and nonabnormal correlated users to find potential abnormal groups in microblog.

6.4. Experimental Analysis. To test the performance of PMCAU in detecting abnormal correlated users and probabilistic reasoning of possible associations between abnormal correlated users, a comparative experiment was set up. DBSCAN-based clustering algorithm and PageRank-based abnormal detection algorithm of microblog account were compared with PMCAU. The detection performance of the three algorithms was compared by corresponding indexes of the experiment.

The DBSCAN-based clustering algorithm is an anomaly detection method based on density clustering, which can find abnormal points while clustering. The PageRank-based microblog account anomaly detection algorithm constructs a social relationship matrix according to the user relationships and ranks the account by iteratively calculating the PageRank value to detect malicious users. The two algorithms have good results in user anomaly detection, so the above two algorithms are selected for comparison experiments with PMCAU. PMCAU is divided into two parts: behavioral correlation analysis and abnormal correlation probabilistic reasoning. Behavioral correlation analysis detects and determines abnormal correlated users among malicious users by calculating their attribute similarity and behavioral interaction. Abnormal correlation probabilistic reasoning is used to analyze and predict the abnormal correlated users to get their correlation probabilistic.

Using these three algorithms, 10 groups of experiments were conducted on the data set of "data1–data10" in turn, which was recorded as "G1–G10." Pre, Rec, and $F1$_score were used as the evaluation criteria of the experiment, and the experimental results are shown in Figure 5.

The results show that the DBSCAN-based clustering algorithm clustering by calculating the Euclidean distance of each data point, because the calculation is numerical attributes, does not consider the user published blog content and other text-based information, and user interaction behavior and other behavioral information, so the Pre of detection is low. Although the PageRank-based abnormal detection algorithm for microblog account has a high Pre, it

Table 2: Symbol description.

Detection result	Actual situation	
	Abnormal correlated users	Nonabnormal correlated users
Abnormal correlated users	TP	FP
Nonabnormal correlated users	FN	TN

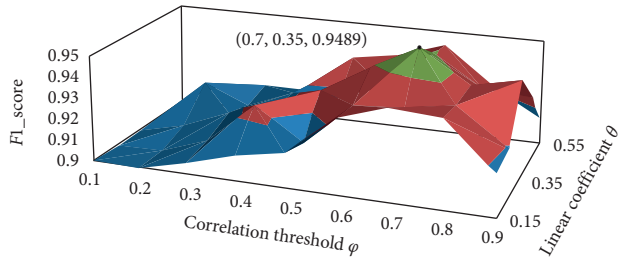


Figure 2: *F*1_score value under different parameters.

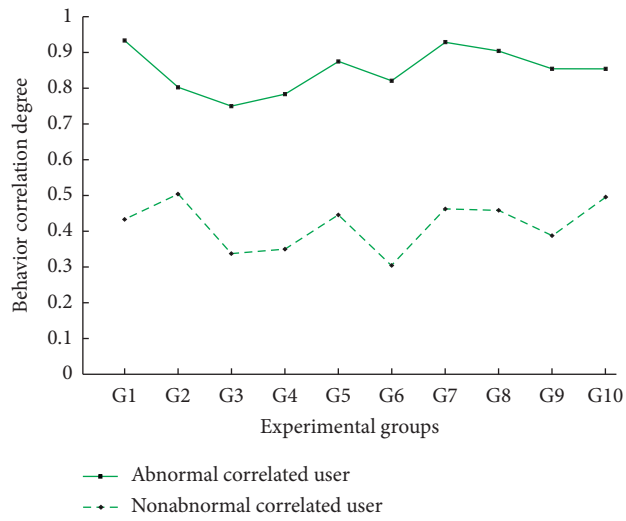


Figure 3: User behavior correlation degree distribution.

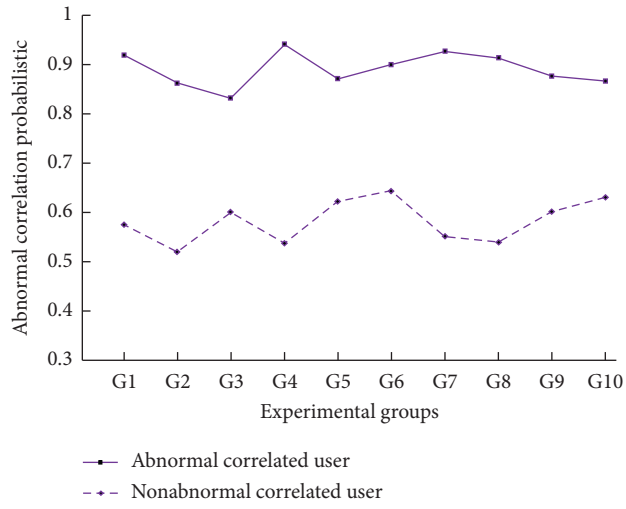


Figure 4: User abnormal correlation probabilistic.

only considers the social relationship between users and does not consider the attributes and features of users, and the ranking results of users are affected by a time delay, so the Rec and *F*1_score are lower.

When detecting abnormal correlated users and inferring abnormal correlation probabilities, PMCAU considers the interaction behavior between users and determines the abnormal correlation users by comprehensive numerical attributes, text type information, and interactive behaviors. Probabilistic soft logic reasoning is correlated with the probabilistic of occurrence of correlation behavior, so it has high Pre, Rec, and *F*1_score.

In order to check the stability of PMCAU, the average and variance of 10 groups of experimental results corresponding to the three algorithms were compared. The experimental results are shown in Figure 6.

It can be observed in Figure 6 that the 10 sets of experiments corresponding to the three algorithms are compared in terms of Pre, Rec, and *F*1_score. Among them, the average value of the three indexes of DBSCAN clustering algorithm is medium. In PageRank ranking algorithm, although the average Pre is 90%, its Rec is low, and the overall performance of the algorithm is medium. The Pre, Rec, and *F*1_score of the 10 groups of experiments corresponding to PMCAU are highest, and the model performance is outstanding, with the Pre reaching 97.35%.

As can be seen from Figure 7, the variance of DBSCAN clustering algorithm and of PageRank ranking algorithm on the four experimental evaluation indexes are large, indicating that the experimental results of the above two algorithms fluctuate greatly in the 10 groups of experiments respectively and the stability of the algorithm is poor. The variance of PMCAU corresponding to the three kinds of experimental evaluation indexes is small, indicating that the experimental results of the corresponding 10 groups fluctuate less and the algorithm is stable.

According to the average and variance of 10 groups of experimental results corresponding to each of the three algorithms, compared with the other three algorithms, PMCAU has good stability and adaptability when it comes to detecting abnormal correlated users of microblog and reasoning abnormal correlation probabilistic while guaranteeing accuracy.

6.5. Reasoning Performance Analysis. To compare the performance difference between PMCAU and the advanced reasoning models in existing research studies, a comparison experiment based on the Markov logic network (MLN) and PMCAU was set up to analyze the reasoning ability of two algorithms for abnormal correlation probabilistic.

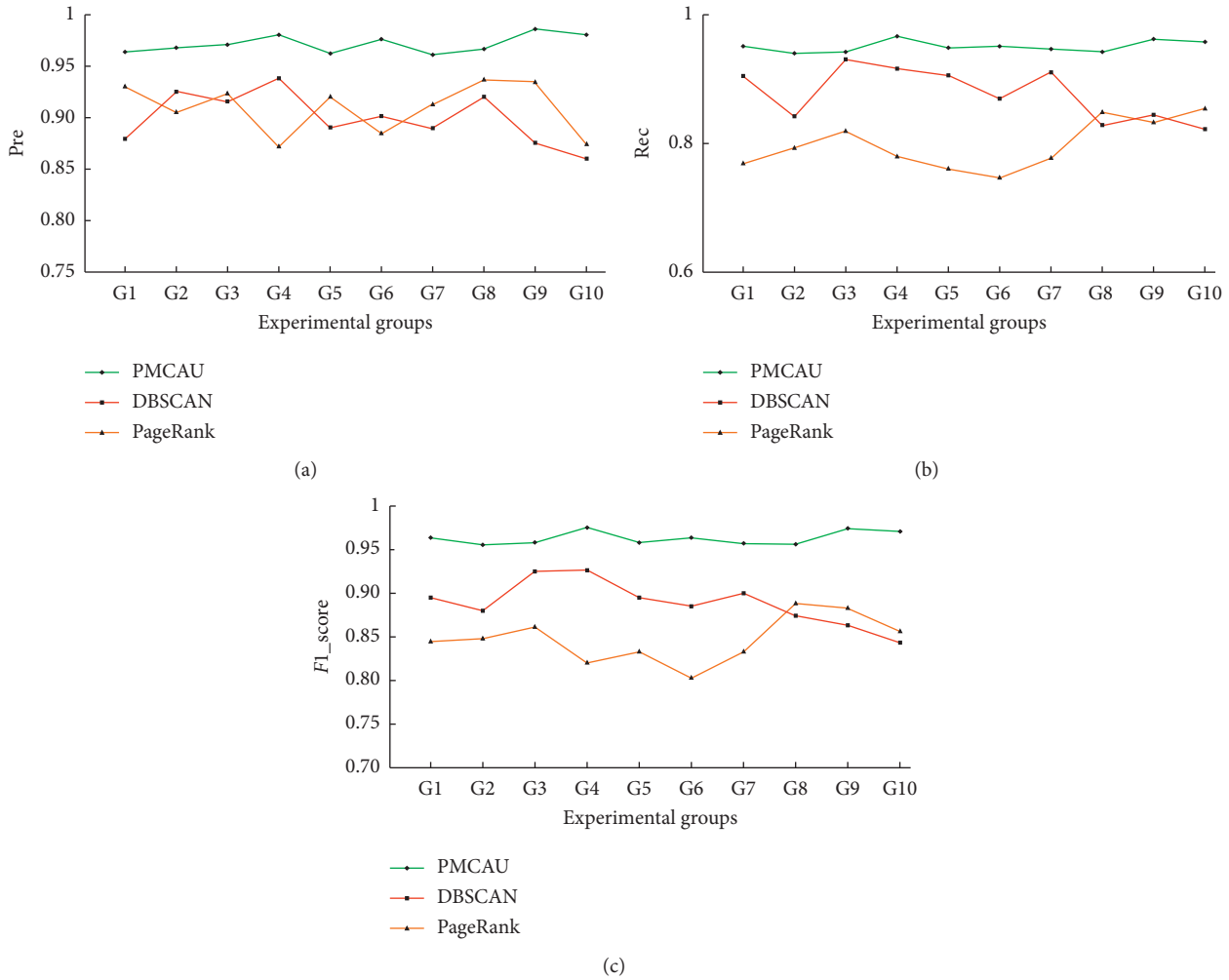


Figure 5: (a) Precision rate. (b) Recall rate. (c) *F*1_score.

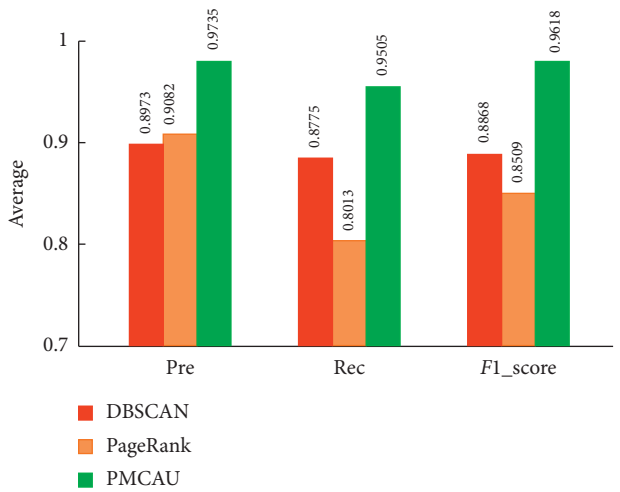


Figure 6: Average.

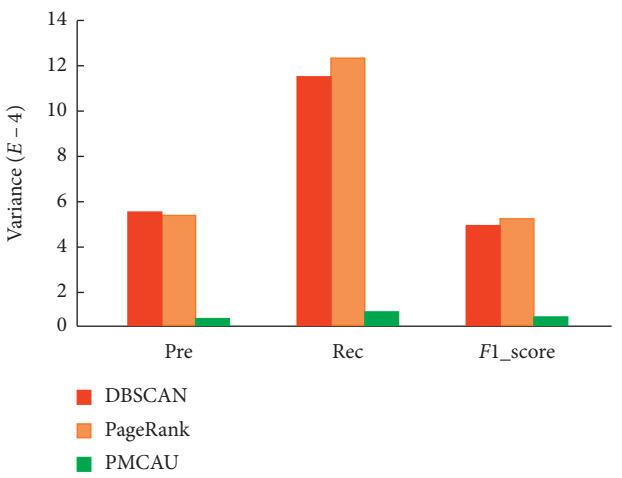


Figure 7: Variance.

Similarly, data groups "Data1–Data10" were taken as data samples for 10 groups of experiments, denoted as "G1–G10." MLN with the same logical reasoning ability was used for the experiment, and the experimental results were compared with PMCAU. During the experiment, 80% of user data of each group of samples were selected to train PMCAU and MLN models, and 20% of user data were taken as test samples. With *F*1_score as the evaluation index, experimental results are shown in Figure 8.

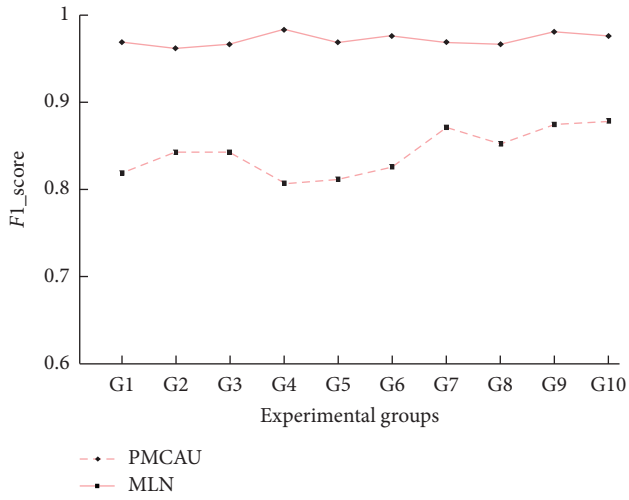


Figure 8: Comparison of $F1$_score between MLN and PMCAU.

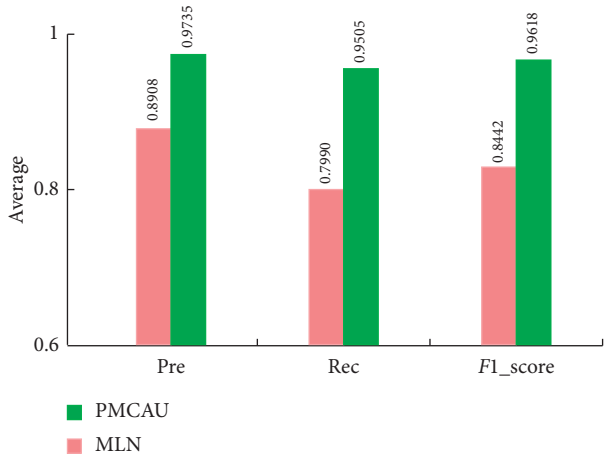


Figure 9: Average of MLN and PMCAU.

The results show that when MLN is used to infer the abnormal correlation probabilistic between abnormal correlated users in microblog, the Boolean value {0, 1} is used to determine whether there is an abnormal correlation, and the complex relationship between users in microblog cannot be better expressed, the corresponding $F1$_score in [0.8, 0.9]. PMCAU uses PSL reasoning—the use of [0, 1] between continuous soft true value as operation values—which makes the reasoning become a continuous convex optimization problem and can effectively express the cause and effect of user behavior and overcome the user's existence time. The limitations of factors such as space and incomplete records, the corresponding $F1$_score is higher, and $F1$_score is stable at [0.9, 1].

The average and variance of the 10 sets of experiments corresponding to MLN and PMCAU are shown in Figures 9 and 10. Among them, although MLN can perform probabilistic reasoning on users' complex relationships to a certain extent, its Pre, Rec, and $F1$_score evaluation standards are all lower than that of PMCAU. Experiments show that PMCAU has better logical reasoning ability and is more suitable for

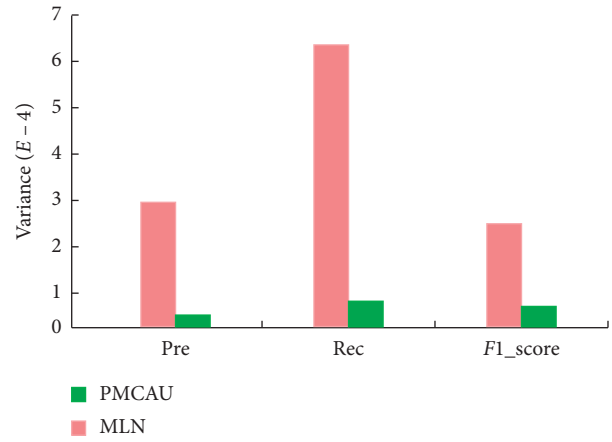


Figure 10: Variance of MLN and PMCAU.

predicting the abnormal correlation probabilistic of abnormal correlated users in a microblog network and identifying potential abnormal user groups under the premise of ensuring accuracy.

7. Conclusion

Aiming at the problem of abnormal user group detection in the IoT, a PMCAU model for abnormal user correlation probabilistic inference is proposed. First, the concept of user behavior correlation degree is proposed for malicious user nodes in the IoT. The similarity measurement algorithm is applied to construct the correlation measurement algorithm. Attribute similarity and behavior interaction were calculated respectively, and user behavior interaction was determined to determine *acu*. Second, probabilistic soft logic is used to express the causal relationship among abnormal user groups. Combined with geographical location, behavioral information, interest preference, and user follower information, a set of reasoning rules more suitable for predicting the possibility of abnormal correlation with users is proposed, and the possibility of abnormal correlation is predicted to determine the potential abnormal group nodes in the IoT.

The experimental results show that PMCAU has high detection accuracy and beneficial stability, and it has a good effect on the correlation probabilistic inference prediction between abnormal behaviors. In the future, the behavior of abnormal correlated users in the IoT will be specifically analyzed, the types of abnormal behaviors will be determined, and the tracking of abnormal behaviors will be further discussed based on the medical IoT.

Data Availability

The data came from an article [15] by Zhang Jing of Tsinghua University, in which crawlers were used to construct a data set of microblog users. The microblogging network they used in this study was crawled from Sina Weibo.com, which, similar to Twitter, allows users to follow each other. Particularly, when user A follows B, B's activities such as tweet

and retweet will be visible to A. A can then choose to retweet a microblog that was tweeted (or retweeted) by B. User A is also called the follower of B, and B is called the followee of A. After crawling the network structure, for each one in the 1,787,443 core users, the crawler collected her 1,000 most recent microblogs. At the end of the crawling, they produced in total 4 billion following relationships among them, with average 200 followees per user.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant no. 2017YFB0802300.

References

[1] K. Fan, S. Sun, Z. Yan, Q. Pan, H. Li, and Y. Yang, "A blockchain-based clock synchronization scheme in IoT," *Future Generation Computer Systems*, vol. 101, pp. 524–533, 2019.

[2] K. Fan, W. Jiang, L. Qi, H. Li, and Y. Yang, "Cloud-based RFID mutual authentication scheme for efficient privacy preserving in IoV," *Journal of the Franklin Institute*, 2019.

[3] Z. Ding, Y. Jia, and B. Zhou, "Survey of data mining for microblogs," *Journal of Computer Research and Development*, vol. 51, no. 4, pp. 691–706, 2014.

[4] Y. Zhou, "Behavior analysis based traffic anomaly detection and correlation analysis for communication networks," Doctoral dissertation University of Electronic Science and Technology of China, Chengdu, China, 2013.

[5] J. Zhao, H. Huang, and S. Tian, "Protocol anomaly detection based on hidden Markov model," *Journal of Computer Research and Development*, vol. 47, no. 4, pp. 621–627, 2010.

[6] X. Xiao and Q. Zhai, "Masqerade detection based on shell commands and high-order Markov chain models," *Acta Electronica Sinica*, vol. 39, no. 5, pp. 1199–1204, 2011.

[7] J. Cao and J. Wu, "Sina microblog information diffusion analysis and prediction," *Chinese Journal of Computers*, vol. 37, no. 4, pp. 779–790, 2014.

[8] Z. H. Xu, Y. Zhang, Y. Wu, and Q. Yang, "Modeling user posting behavior on social media," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Portland, OR, USA, August 2012.

[9] S. Li, X. Li, and H. Yang, "A zombie account detection method in microblog based on the pagerank," in *Proceedings of the 2017 IEEE International Conference on Software Quality, Reliability and Security Companion*, Prague, Czech Republic, July 2017.

[10] C.-F. Xu, C.-L. Hao, B.-J. Su, and J.-J. Lou, "Research on Markov logic networks," *Journal of Software*, vol. 22, no. 8, pp. 1699–1713, 2011.

[11] C. Sun, Z. Yan, Q. Li, Y. Zheng, X. Lu, and L. Cui, "Abnormal group-based joint medical fraud detection," *IEEE Access*, vol. 7, pp. 13589–13596, 2018.

[12] Z. Yang, Q. Sun, and B. Zhang, "Evaluating prediction error for anomaly detection by exploiting matrix factorization in rating systems," *IEEE Access*, vol. 6, pp. 50014–50029, 2018.

[13] A. Kimmig, S. Bach, and M. Broecheler, "A short introduction to probabilistic soft logic," in *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, Lake Tahoe, NV, USA, 2012.

[14] S. Tomkins, J. Pujara, and L. Getoor, "Disambiguating energy disaggregation: a collective probabilistic approach," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2857–2863, Melbourne, Australia, August 2017.

[15] J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing, "Who influenced you? Predicting retweet via social influence locality," *Acm Transactions on Knowledge Discovery from Data*, vol. 9, no. 3, pp. 1–26, 2015.

[16] M. Yang, J. M. Yin, and G. L. Ji, "Classification methods on imbalanced data: a survey," *Journal of Nanjing Normal University (Engineering and Technology Edition)*, vol. 8, no. 4, pp. 7–12, 2008.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 4960964, 9 pages
https://doi.org/10.1155/2020/4960964

Research Article

Preserving Data Privacy in the Internet of Medical Things Using Dual Signature ECDSA

Maria-Dolores Cano and Antonio Cañavate-Sanchez

Department of Information and Communication Technologies, Universidad Politecnica de Cartagena, Cartagena 30202, Spain

Correspondence should be addressed to Maria-Dolores Cano; mdolores.cano@upct.es

Received 25 October 2019; Revised 1 February 2020; Accepted 17 February 2020; Published 10 June 2020

Guest Editor: Kewei Sha

The disclosure of personal and private information is one of the main challenges of the Internet of Medical Things (IoMT). Most IoMT-based services, applications, and platforms follow a common architecture where wearables or other medical devices capture data that are forwarded to the cloud. In this scenario, edge computing brings new opportunities to enhance the operation of IoMT. However, despite the benefits, the inherent characteristics of edge computing require countermeasures to address the security and privacy issues that IoMT gives rise to. The restrictions of IoT devices in terms of battery, memory, hardware resources, or computing capabilities have led to a common agreement for the use of elliptic curve cryptography (ECC) with hardware or software implementations. As an example, the elliptic curve digital signature algorithm (ECDSA) is widely used by IoT devices to compute digital signatures. On the other hand, it is well known that dual signature has been an effective method to provide consumer privacy in classic e-commerce services. This article joins both approaches. It presents a novel solution to enhanced security and the preservation of data privacy in communications between IoMT devices and the cloud via edge computing devices. While data source anonymity is achieved from the cloud perspective, integrity and origin authentication of the collected data is also provided. In addition, computational requirements and complexity are kept to a minimum.

1. Introduction

Our physical universe is acquiring a new digital existence with the arrival of the Internet of Things (IoT). Many beings/objects are expected to have connectivity and the capacity to collaborate. With billions or trillions of IoT devices connecting to the cloud to exchange, process, and store information, the network architecture must adapt in the most agile, intelligent, and efficient way possible to maintain the quality of the provided services while considering the heterogeneity of networks and devices. Despite the advantages of a conventional, centralized cloud model, the future IoT faces significant challenges: latency, velocity, volume of data, location awareness, mobility support, or monopoly versus an open IoT contention, among others [1, 2]. This is of great importance in the Internet of Medical Things, since data are not only used for disease prediction but also for health monitoring and treatment, where it is vital to control these key performance metrics [3–5].

Edge computing can address these challenges by offering the additional computing, storage, and communication resources for particular tasks, thus liberating both IoMT devices and the cloud and improving the performance of traditional cloud computing services [6]. However, one key concern about the use of edge computing is security. The edge not only inherits some of the cloud's security challenges but also attributes to new vulnerabilities and threats (e.g., in terms of secure data computation, secure data storage, privacy protection, authentication, and access control [7]). Particularly, the authors focus this work on how to preserve the privacy of data sent by IoMT devices to the cloud using edge computing while at the same time permitting the cloud and the edge devices to authenticate the integrity and the origin of the data. Authentication is defined as the ability to

demonstrate you are who you say you are. In terms of data exchange in a communication network, there is authentication if the sender of a message can be identified unequivocally by the receiver. In turn, there is integrity if it can be demonstrated that a message/information has not been created, modified, or deleted by unauthorized users or systems.

In this work, the authors propose a method to be used in IoMT scenarios that is able to provide data integrity and data privacy while guaranteeing that the data have come from an authenticated IoMT source. To this end, the authors introduce the concept of dual signature (DS) in the elliptic curve digital signature algorithm (ECDSA) [8]. Note that a dual signature is not a double signature, but a technique to couple two values of different natures, keeping them anonymous to two different entities in a secure fashion. Besides simplicity, the authors' approach differs from previous works in that it is compatible with hardware implementations. Recent works have demonstrated that public key cryptography with elliptic curve cryptography (ECC) in constrained IoT devices, in general, is not a concern. Furthermore, ECDSA signature creation is affordable and effective [9–11]. Moreover, ECDSA signature verification, which is considered to be a computationally intensive task [12], will not be carried out by IoMT devices but by edge network elements, which have no operational constrains, thus making this an appropriate, agile, and simple solution for IoMT environments.

The rest of the paper is organized as follows. Section 2 reviews the state of the art, showing related works from the scientific literature. In Section 3, the authors introduce the concept of dual signature in ECDSA, describing the communication process from the IoMT transmission device to the cloud via edge computing elements, demonstrating its security features. Section 4 is devoted to security analysis and computational requirements. The paper ends summarizing the most important outcomes.

2. Related Works

It is important to note that providing data privacy in terms of anonymity and integrity is needed not only in advanced health systems but also in other scenarios such as intelligent traffic systems (ITS) dealing with driver or vehicle information or in collaborative social applications managing peoples' data. Therefore, it is encouraging to observe the proposals that researchers are suggesting in these other communication fields. In this regard, several works can be found in the related literature addressing the preservation of data privacy in IoT [13–20].

In [14], the authors presented a public key ECC-based solution for intelligent transportation environments, where the task of authenticating the vehicles within the coverage of a road side unit (RSU) was a shared assignment between the vehicles themselves and the RSU. Specifically, those vehicles with better computation resources and which were closer to the RSU were selected as edge nodes. These vehicles were then responsible for the authentication of messages transmitted by nearby vehicles, incorporating batch authentication. They were also responsible for sending the results to the RSU, which then verified the previously processed information. The authors also proposed the use of a cuckoo filter and fuzzy logic to speed up the process. It is important to note that in [14], there are third-party authorities that are trusted by all entities (one for each RSU), which are able to ascertain the real identity of the vehicles. A similar approach is followed in [18]. In [15], several Bloom filter probabilistic data structures are employed to authenticate both vehicles and unmanned aerial vehicles (UAV). Basically, the IDs of vehicles under UAV coverage that have been authenticated are hashed and stored in Bloom filters, and thus messages from these vehicles are only forwarded to the next communication element if the UAV queries the Bloom filters and the result is positive. No more information about the authentication, integrity, or privacy processes was provided in that work.

Li et al. introduced in [16] a homomorphic Boneh–Goh–Nissim-based method for preserving privacy in mobile edge computing scenarios. The solution seems to be very interesting and robust from a security perspective. The performance evaluation of this method was previously presented in [21]. Similar approaches to [16, 21] were proposed by Wang et al. [22] and Wang [23]. In both cases, the proposals were based on the use of homomorphic encryption to provide confidentiality. In the former, privacy was achieved by using pseudonyms when data are forwarded from the edge/fog computing device to the cloud, instead of using the device identification information. Aggregation at the edge/fog device allowed for a more efficient data transmission to the cloud in terms of overhead compared to other methods, as shown by the authors. In the latter, the same idea of including an intermediate element (edge or fog device) to aggregate data and to provide users' privacy is proposed, with comparable results. However, it is noteworthy to mention that possible limitations to the use of homomorphic encryption could arise in terms of IoT device energy consumption. Nevertheless, these challenges could be reduced or even resolved as new improvements are incorporated into homomorphic encryption techniques, as indicated in [24].

Particularly for the IoMT paradigm, its novelty limits the contributions found in the scientific literature. Deebak et al. presented in [25] an anonymous and secure user authentication method based on biometric data to protect communications in healthcare applications. Their proposal was also based on the use of elliptic cryptography, together with smart cards that stored users' biometric information. Once a user was authenticated, a key generation process started so that the communication channel would be made secure (ciphered) using this key. Two possible limitations of this proposal are the necessity of using physical smart cards (an active approach from the users' perspective) and the congestion that could appear in case of a high number of IoMT devices, as the authors state in their conclusions.

In [26], the authors proposed a novel method for encryption and encoding to be used in IoMT based on the Advanced Encryption Standard (AES). They experimentally tested the performance of their proposal, whose main

advantage was that the time required to perform the encryption and encoding processes was shorter compared with traditional cryptographic techniques. As another example, the authors in [27] proposed a key generation mechanism using biometric information as input. The keys were then employed for medical data encryption. As a key generation method, their proposal outperformed other existing technologies.

From a different perspective, Guan et al. addressed in [28] privacy in IoMT by using machine learning. Their goal was to guarantee that by accessing the medical information dataset, an attacker could not obtain specific individual information but only approximate data. In order to do so, they suggested an original process to update the centroids of the clusters, which are used for clustering-based learning, incorporating controlled noise. The results were notable, but as indicated by the authors, there is a trade-off between privacy preservation and the accuracy of cluster results. Other works can be found dealing with the assessment of security levels in IoMT [29, 30] or how to perform accurate auditing actions [31].

The approach introduced in this paper differs from previous works in two main factors: simplicity and hardware compatibility. Although Bloom filters and other more recent data structures such as cuckoo filters are very promising for security applications, they still face problems having to do with hardware implementation [32]. Nevertheless, it is important to observe that our proposal is compatible with the use of these membership query techniques. In addition, previous works have mostly focused on how to achieve a successful level of confidentiality by improving either the encryption technique or the key generation process. In this work, our proposal is not focused only on confidentiality but also on how to protect the anonymity of the person/device that generates the data, with the awareness that data confidentiality can be added as another security layer depending on the energy and computational restrictions of the IoMT source device.

3. ECDSA with Dual Signature

3.1. System Description. Digital signatures have been widely used since their introduction in cryptosystems [33]. Dual signature was presented in [34] as an effective way to link two different types of information in e-commerce, particularly, the buyer's order information (OI) and the buyer's payment information (PI). Linking is done in such a way that the PI is hidden from the seller and the OI is hidden from the bank, but both recipients (seller and bank) can unquestionably verify the authenticity and integrity of both data. Dual signature can be implemented with any asymmetric encryption algorithm.

Figure 1 shows the general procedure of a dual signature. As depicted in Figure 1(a), both the OI and PI are individually hashed. Then, these two hashes are concatenated and hashed. The resulting hash is encrypted with the client's private key and the output is called a dual signature. Observe that when the client sends a message to the seller and the bank (Figure 1(b)), the seller receives the OI in plaintext and the hash of the PI. Therefore, the seller can verify the dual signature without receiving the payment information and using the client's public key. The same applies to the bank, but in this case, the information that the seller forwards to the bank is only what appears encrypted with the bank's public key (K_{PBank}) in Figure 1(b). Consequently, the bank will not know what the client bought (the OI) and will only know the payment information.

The authors' proposal inherits the procedure shown in Figure 1 and adapts it to the IoMT paradigm. Figure 2 represents a general IoT communication scenario with three participants, namely, transmission devices (TDs), edge computing servers/devices (ECSs), and the cloud (C). TDs are IoT devices with computational and energy constraints that collect and send data to the C via an ECS. ECSs are located near TDs, at the edge of the network, and they have computing abilities. Smartphones or computers can be examples of ECS devices. C is a central cloud service that stores and processes data.

Table 1 includes all the notations that will be used hereinafter. The proposal is based on the use of ECC [35, 36]. It is assumed that all participants go through a secure initiation phase to obtain a private/public ECC key pair (d, Q), using G as the generator point of the elliptic group $E_p(a, b)$ and n being a very large integer. Alternatively, the key pairs (d, Q) could be obtained using a prestored strategy. In any case, private keys are kept secret and the relationship between private and public keys is $Q = d \cdot G$.

Once key pairs are generated, C's public key Q_C is published and veritably known by all TD_i and ECS_j, where $i = \{1, 2, \ldots, m\}$, $j = \{1, 2, \ldots, z\}$, and $z << m$. Likewise, each ECS_j knows the public keys Q_{TDi} of all TD_i under its coverage. Note that C does not need to be aware of TD_i's public keys. Then, when an IoMT device TD_i has collected information m that needs to be sent, it proceeds as follows:

(1) TD_i selects a random (or pseudorandom) integer k, $k \in [1, n-1]$.

(2) TD_i computes $P_1(x_1, y_1) = k \cdot G$ and r is defined as follows

$$r = x_1 \bmod n. \tag{1}$$

(3) Then, TD_i computes $e = H(m)$, $f = H(ID_{TDi})$, and $g = H(e \,||\, f)$. In all cases, H should be a strong hash function (e.g., SHA-2 or SHA-3)

(4) Finally, TD_i calculates s as shown in equation (2). The obtained dual signature is the pair (r, s).

$$s = k - 1\,(g + d_{TDi} \cdot r) \bmod n. \tag{2}$$

At this point, TD_i sends a message M_1 to ECS_j containing health-related data. M_1 is depicted in Figure 3. This message M_1 has two parts. The first part $\{ID_{TDi}, e, (r, s)\}$ is sent in plaintext and contains the following information: the identification of TD_i, the hash e of the collected health data m, and the dual signature (r, s). The second part of M_1 is

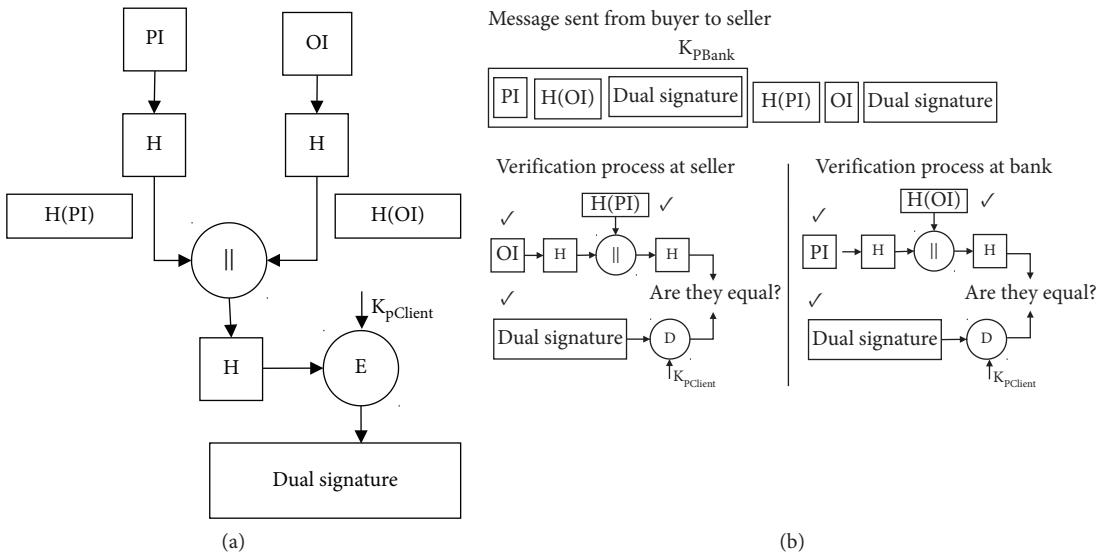


Figure 1: General procedure of a dual signature where H represents a secure hash function (E≡encrypt; D≡decrypt; ||≡concatenate; $K_{pclient}$≡the buyer's private key; $K_{Pclient}$≡the buyer's public key; K_{PBank}≡the bank's public key; ✓≡available data): (a) dual signature generation and (b) message sent from the buyer to the merchant and to the bank together with the dual signature verification processes.

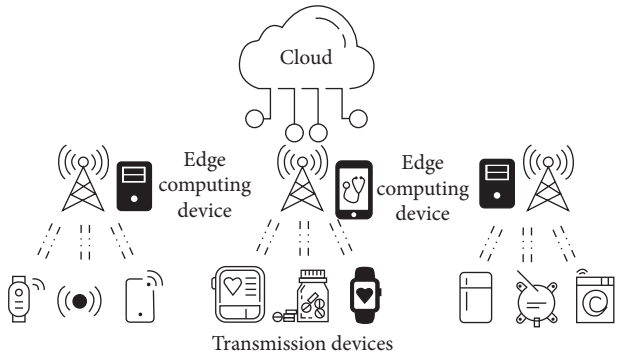


Figure 2: IoT communication scenario.

Table 1: Notation.

Symbol	Meaning
TD_i	The i-th transmission device
ECS_j	The j-th edge computing device/server
ID_{TDi}	The identification of TD_i
G, n	A generator point in $E_p(a, b)$ with a very large order n
H	A secure hash algorithm
d_{TDi}	The private key of TD_i
Q_{TDi}	$Q_{TDi} = d_{TDi} G$, the public key of TD_i
d_C	The private key of C
Q_C	$Q_C = d_C \cdot G$, the public key of C
(r, s)	The dual signature
P	Points within $E_p(a, b)$
P	A prime number
E	The hash of data m
F	The hash of the value ID_{TDi}
G	$H(e \| f)$, the hash of e and f
M	Data to be sent by a transmission device TD_i

encrypted with an asymmetric cryptographic technique using the public key of the cloud, Q_C. Any asymmetric encryption technique can be used depending on the

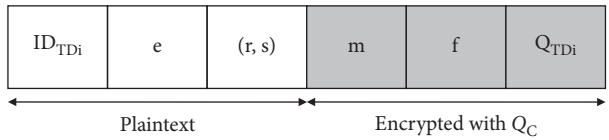


Figure 3: Message M_1 from TD_i to ECS_j.

capabilities of TD_i. The encrypted data that M_1 contains are the collected health data m, the hash f of the identification of TD_i, the public key Q_{TDi}, and the dual signature (r, s).

Upon the reception of M_1, the edge device ECS_j verifies the authenticity and integrity of M_1 using the public key Q_{TDi} as follows:

(1) ECS_j verifies that both r and s are integers, i.e., $(s, r) \in [1, n-1]$.

(2) ECS_j calculates $f = H(ID_{TDi})$ and $g = H(e \| f)$; observe that ID_{TDi} and e were sent as plaintext in M_1 (Figure 3).

(3) Then, the ECS_j calculates w as shown in the following equation:

$$w = s^{-1} \bmod n. \tag{3}$$

(4) It calculates u_1 and u_2 as depicted in equations (4) and (5):

$$u_1 = w \cdot g, \tag{4}$$

$$u_2 = w \cdot r. \tag{5}$$

(5) From u_1 and u_2, ECS_j computes the point P_2 as shown in equation (6). Observe that, as usual in

asymmetric methods, ECS_j knows the public key of TD_i.

$$P_2(x_2, y_2) = u_1 \cdot G + u_2 \cdot Q_{TDi}. \quad (6)$$

(6) Then, ECS_j computes $v = x_2 \bmod n$.

Consequently, if $v = r$, then ECS_j accepts the dual signature, or else it rejects it. Even though ECS_j does not have access to the collected health data m (note that m is encrypted with Q_C as depicted in Figure 3), ECS_j can guarantee that TD_i was the IoMT device that sent this information m. The reason is that only TD_i knows its secret key d_{TDi}, which was used to create the dual signature. In addition, ECS_j knows that m has not been modified, hence confirming the integrity of the information; otherwise, the dual signature would have been invalid (and rejected). The demonstration of the verification of the dual signature is detailed later in Section 3.2.

Next, we assume that ECS_j sends a message M_2 to C. The message M_2 also has two parts, as illustrated in Figure 4. The first part will be used by C to authenticate the source of this message. This could be done with a classic ECDSA signature. In Figure 4, ID_{ECSj} is the ID of ECS_j, which sends this message, and h is the resulting hash of the complete message M_2. The second part of M_2 is equal to the batch of all the encrypted data in messages M_{1i} coming from the different IoMT devices TD_i within the coverage of the same ECS_j. In other words, ECS_j appends each grey part corresponding to the encrypted information that each TD_i transmitted to ECS_j $\{m, f, Q_{TDi}, (r, s)\}_{Qc}$. This message M_2 is sent from ECS_j to C. Upon the arrival of M_2 to the cloud C and after verifying the origin and integrity of this message by checking the ECDSA classic signature, C proceeds as follows:

(1) C decrypts all blocks $\{m, f, Q_{TDi}, (r, s)\}_{Qc}$ using the cloud's private key d_C.

(2) For each block, C calculates $e = H(m)$ and $g = H(e \| f)$.

(3) Then, it calculates $w = s^{-1} \bmod n$.

(4) Now, C calculates u_1 and u_2 as depicted in equations (7) and (8):

$$u_1 = w \cdot g, \quad (7)$$

$$u_2 = w \cdot r. \quad (8)$$

(5) C computes the point as P_3 as indicated in the following equation:

$$P_3(x_3, y_3) = u_1 \cdot G + u_2 \cdot Q_{TDi}. \quad (9)$$

(6) Finally, C computes $v = x_3 \bmod n$.

As described before, if $v = r$, the dual signature is accepted by the cloud C (otherwise, it is rejected). After this operation, C can guarantee that the received data m has not been modified and that m was sent by an authenticated IoMT TD, although the identity of this device is unknown to C. Observe that C knows the value of the public key Q_{TDi}, but it does not know the identity of TD_i. In other words, health data privacy is preserved without losing origin authentication and integrity.

3.2. Demonstration. In order to demonstrate the goodness of the proposal, let us assume that ECS_j has received the message $M_1 = \{ID_{TDi}, e, \{m, f, Q_{TDi}\}_{Qc}, (r, s)\}$. Let us also assume that M_1 has not been altered. Then, from equation (2) we can carry out the following operations:

$$\begin{aligned} k &= s^{-1}(g + d_{TDi \cdot r}) \bmod n, \\ k &= \left(s^{-1} \cdot g + s^{-1} \cdot d_{TDi \cdot r}\right) \bmod n. \end{aligned} \quad (10)$$

In equation (10), we can substitute some terms using equations (3)–(5), so the new expression will be

$$\begin{aligned} k &= (w \cdot g + w \cdot d_{TDi \cdot r}) \bmod n, \\ k &= (u_1 + u_2 \cdot d_{TDi}) \bmod n. \end{aligned} \quad (11)$$

At the transmission device TD_i we defined $P_1(x_1, y_1) = k \cdot G$, whereas in reception (at the ECS_j), we have that $P_2(x_2, y_2) = u_1 \cdot G + u_2 \cdot Q_{TDi}$. If P_1 is equal to P_2, then r and $v$would be equal and the dual signature would be correct because both values r and v correspond to the x coordinates of P_1 and P_2, respectively. Let us verify this by taking into account that the public key of TD_i was obtained as $Q_{TDi} = d_{TDi} \cdot G$:

$$\begin{aligned} P_2(x_2, y_2) &= u_1 \cdot G + u_2 \cdot Q_{TDi} = u_1 \cdot G + u_2 \cdot d_{TDi} \cdot \\ G &= (u_1 + u_2 \cdot d_{TDi}) \cdot G. \end{aligned} \quad (12)$$

Subsequently, applying equation (11), we have that

$$P_2(x_2, y_2) = (u_1 + u_2 \cdot d_{TDi}) \cdot G = k \cdot G = P_1(x_1, y_1). \quad (13)$$

Accordingly, both values $r = x_1 \bmod n$ (calculated at TD_i) and $v = x_2 \bmod n$ (calculated at ECS_j) will be equal. Any modification of the transmitted values in M_1 would cause different values for e or f and therefore for g, leading to the detection of the attack. The same demonstration procedure should be applied for M_2.

4. Security Analysis

The security characteristics of the proposal are analyzed in this section, demonstrating that it complies with the stated security requirements for IoMT scenarios.

4.1. Message Authentication. The legitimacy of the sender of a message is guaranteed by the digital signature ECDSA. The secret key d_{TDi} is only known by the transmission device TD_i. This secret value is employed to compute the digital signature as shown in equation (2). Assuming that TD_i was resistant to tampering, this key could not be retrieved by an attacker. Accordingly, TD_i could not be impersonated since

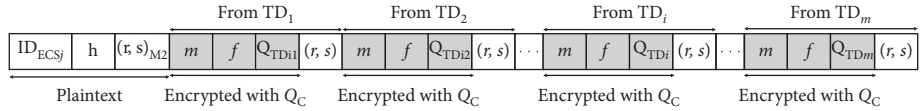


Figure 4: Message M_2 from ECS_j to C.

an attacker would not be able to generate a valid digital signature.

For instance, let us assume that an attacker modifies ID_{TDi} in message M_1 (Figure 3), attempting to impersonate TD_i. Then, the corresponding hash f' would be different from f, so $g = H(e||f')$ would also be different than g, and the digital signature verification would be detected as nonvalid.

4.2. Identity Privacy. The proposed dual signature procedure guarantees data privacy as follows: (i) health data sent by the transmission devices are hidden from the edge device, but not the identifiers, and (ii) the identities of the transmission devices are hidden from the cloud, but not the health data.

The identity of a transmission device TD_i is only known by ECS_j. Indeed, ECS_j receives the identification of each TD_i that sends a message of type M_1 (as depicted in Figure 3). The reason for allowing the ECS to be aware of the identity of the transmission devices is that the former needs to associate the identity of TD_i to the corresponding public key Q_{TDi} to verify the digital signature. However, it is important to realize that ECS_j does not know the information m that TD_i is sending to the cloud: information m is kept secret from the ECS_j.

On the other hand, when C receives messages of type M_2 (see Figure 4) from an ECS_j, the cloud cannot deduce the identity of the TD_i that sent that information because C only knows the hash of ID_{TDi}, which is irreversible if a strong hash function has been used. Observe that C will need to be able to verify the public key of ECS_j, so the identity of ECS_j is not hidden from C.

4.3. Data Tampering. The use of strong hash functions guarantees integrity and security against data tampering. In the communication part from TD_i to ECS_j, if an attacker alters ID_{TDi}, e, or the digital signature itself (r, s) in M_1 (see Figure 3), the verification process would detect the attack because the resulting hashes would be different; therefore, the verification would be erroneous, resulting in the rejection of the digital signature.

An attacker could also try to modify the encrypted part of M_1 (Figure 3). The procedure would be as follows. The attacker captures M_1. Then, it maintains the first part of the message unaltered (the one that is in plaintext), but it creates fake values for m and f and provides a false key $Q_{TDi'}$. However, when the digital signature from TD_i is checked at the cloud C, this digital signature is detected as invalid. Another option for the attacker would be to modify the encrypted part of M_2 (Figure 4): any part of the batched messages from the TDs. But in this case, the verification of the ECDSA signature introduced by the ECS_j in M_2 (as shown in Figure 4) would detect the attack.

4.4. Replay Attacks. In order to avoid attacks in which messages are captured by an attacker and later injected/replayed into the network, timestamps or sequence numbers could be used. If a TD_i sends a timestamp together with the data m, then the ECS_j could verify whether the message has expired (e.g., assuming that data have a validity time of x units of time) and if so, reject the message. Using sequence numbers, the ECS_j could also verify that this number is not repeated within a transmission window. We have not included the use of timestamps or sequence numbers in this paper to provide a clearer understanding of the proposal.

5. Performance Evaluation

In this section, we consider the computational cost and the communication cost of the dual signature ECDSA, introduced in this paper. We also compare the performance with other related schemes. Particularly, we focus on using $E_p(a, b)$ with p of a length of 256 bits. By doing so, the security level would be equivalent to using RSA with an N length of approximately 3000 bits. The selected hash function is SHA-256.

5.1. Computational Cost. For this evaluation, it is assumed that IDs will have a length of 32 bits (4 bytes), and messages will have a size of 1024 bits (128 bytes). We also assume that the IoMT scenario has m transmission devices TD_i, where $i = \{1, 2, \ldots, m\}$, and z edge devices ECS_j, where $j = \{1, 2, \ldots, z\}$ and $z << m$. Then, in order to study the computational cost of this proposal, the times required for performing the most relevant operations will be taken into account as indicated in [37, 38], the latter using an Intel Xeon CPU (E3-1220 V2) at 3.10 GHz in 64 bit mode and the GCC 5.4.0 compiler. It is important to note that these times will vary notably depending on the platforms where the algorithms are run. Numerous works from the related literature can be found addressing improvements in the execution times of ECC cryptographic operations [12, 39].

Observe that to generate message M_1 (Figure 3), a TD_i needs

(a) To generate three hashes, namely, e, f, and g

(b) To encrypt the message m, the hash f, and the public key Q_{TDi}

(c) To generate the digital signature (r, s) with ECDSA

Table 2 details the notation and time cost of the different cryptographic operations. Taking into account that $e = H(m)$, $f = H(ID_{TDi})$, and $g = H(e||f)$, a TD_i needs to generate three hashes with inputs of 128 bytes (1024 bits), 4 bytes (32 bits), and 64 bytes (512 bits), respectively. Thus, the total cycles required for hashing are $(128 + 4 + 64) \cdot T_{Hash}$. In

Table 2: Notation and time cost (at the transmission devices) of the cryptographic operations used in the comparative performance evaluation. In our proposal, it includes P256 ECC, AES CTR 256, and SHA256 [37, 38].

Symbol	Meaning	Time cost				
		Cryptool [40]	Boneh–Goh–Nissim [21]	Castagnos–Laguillaumie [22]	Homomorphic identity-based method [23]	Our proposal
T_{Sig}	Signature creation	2.88 ms	0.969 ms	0.924 ms	0.629 ms	0.918 ms
T_{Ver}	One signature verification	8.53 ms	14.339 ms	27.974 ms	27.349 ms	26 ms
T_{Hash}	SHA-256	15.8 cycles/byte	5.174 μs/byte	—	—	4.726 μs/byte
T_{Enc}	Time for encryption	18.2 cycles/byte	0.828 ms	0.756 ms	1.098 ms	99.82 μs/byte
T_{TOTAL_TD}	Total time at TD	—	1.7968 ms	29.656 ms	1.727 ms	21.009 ms

addition, if AES in Counter Mode (CTR) is employed to generate the encrypted part of M_1, then the time required for encryption in TD_i would be $(128 + 32 + 32)\cdot T_{Enc}$. Finally, the time required to generate the digital signature ECDSA would be T_{Sig}. Consequently, the total computational cost for each IoMT transmission device TD_i would be $(128 + 4 + 64)\cdot T_{Hash} + (128 + 32 + 32)\cdot T_{Enc} + T_{Sig} \approx 21$ ms.

At the edge device, ECS_j, the time required to verify the digital signature and to batch the health data sent from all the TD_i elements under its coverage (or associated to it) would be the following. Assuming there are x TD_i elements for one ECS_j, then this needs to verify x ECDSA signatures and needs to calculate $2{\cdot}x$ hashes, namely, $f = H(\mathrm{ID}_{\mathrm{TD}i})$ and $g = H(e \,||\, f)$. Consequently, the time required is $x\cdot T_{Ver} + (4 + 64)\cdot T_{Hash}$. If the verification is successful, then the ECS_j batches the encrypted health data received from all its TD_i and carries out two actions to generate M_2. First, it calculates the hash h of the complete message M_2. Second, it creates the digital signature ECDSA of the whole message M_2. Thus, this time corresponds to $((128 + 16 + 32) + 64)\cdot x\cdot T_{Hash}$ cycles plus T_{Sig}. In sum, the total computational cost of verification and aggregation for each ECS_j is $x\cdot T_{Ver} + (4 + 64)\cdot T_{Hash} + ((128 + 16 + 32) + 64)\cdot x\cdot T_{Hash} + T_{Sig} =$. Since the cloud device C is not expected to have computation limitations, the time required to perform the corresponding operations is not included, although its calculation is straightforward. It is also relevant to note that the verification of a digital signature with ECDSA requires a double scalar multiplication on an elliptic curve, and this is an operation with a higher impact in execution time and therefore in energy consumption, as has been demonstrated in the related literature.

Comparing this performance with other relevant schemes, we find out the following. We gather in Table 2 the time cost of all cryptographic operations for several hardware/software configurations as found in the scientific literature. In terms of computation cost for the IoMT devices, the proposal introduced by Li et al. [16] has a total computation cost for each TD_i equal to $2T_{e2} + T_{mp} + T_e$, as indicated by the authors. In particular, $2T_{e2}$ is the time needed to encrypt the health data and $T_{mp} + T_e$ is the time needed for signature creation (see Table 3). Similarly, the method presented in [22] requires $T_e + T_{e2}$ for the cyphering process, $T_{mp} + T_s$ for signature creation, and $2T_p + T_{mp}$ for verification (see Table 3). As another example, the method introduced in [23] requires a total computation cost of $(2T_e + T_{e2}) + (T_{mp} + T_m)$, the former for encryption and the latter for signature creation (see Table 3). As previously mentioned, these times will vary according to the hardware and/or software characteristics of the device that runs these functions. However, if we compare the total computational cost for the TD, we can see in last row of Table 2 that our scheme performs better than [22] and worse than [21, 23]. The reason lies on the fact that we are using AES CTR for encryption, which heavily influences the performance. Nevertheless, observe that the dual signature ECDSA could be compatible with homomorphic-based cryptosystems, avoiding the use of AES and highly reducing the time cost.

Table 3: Notation and time cost of cryptographic operations from [16, 21] and used also in [22, 23].

Symbol	Meaning	Time (ms)
T_{e2}	Time of double exponentiation in a cyclic group	0.4139
T_{mp}	Time of map to point	0.6272
T_e	Time of exponentiation	0.3418
T_p	Time of bilinear pairing	13.6736
T_i	Time of inversion in cyclic group	0.0256
T_m	Time of multiplication in group	0.0019
T_s	Time of scalar multiplication	0.2986

Regarding the performance of the edge device ECS, Figure 5 represents the time cost from the ECS perspective as a function of the number of TD under its coverage. Assuming there are x TD_i elements for one ECS_j, the total time cost for an ECS_j in our proposal is equal to $x\cdot T_{Ver} + (4 + 64)\cdot T_{Hash} + ((128 + 16 + 32) + 64)\cdot x\cdot T_{Hash} + T_{Sig}$. If we substitute the values using Table 2, then the total computational cost is $(x{\cdot}27{,}134) + 1{,}239$ ms. As observed in Figure 5, our scheme is affected by the use of the AES algorithm for encryption, and thus any modification in this task will benefit our proposal. It is important to note that using AES is just an example for encryption, but our proposal does not require to employ this algorithm in order to apply the dual signature ECDSA.

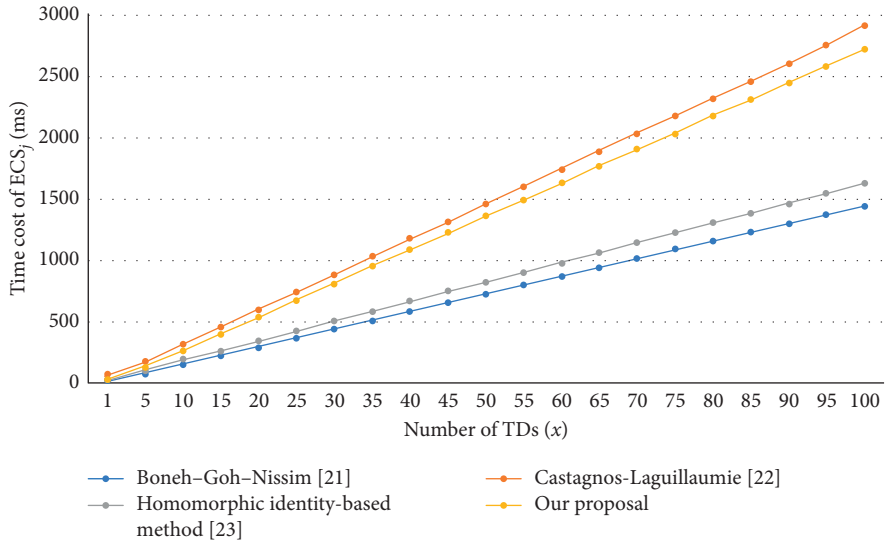


Figure 5: Computational cost for an ECS$_j$.

5.2. Communication Cost. To assess the communication cost, we assume that there are a total of z ECS and that each ECS includes x TD devices. Then, the communication cost would be as follows. The message M_1 sent from each TD$_i$ contains $[\{ID_{TDi}, e, (r, s)\}, \{m, f, Q_{TDi}\}]$, as depicted in Figure 3. Without taking into account the health data m, the communication overhead would be $(4 + 32 + 64 + 32 + 8)$ bytes, respectively, i.e., 140 bytes. Similarly, the message M_2 sent from an ECS$_j$ to the cloud C contains $[\{ID_{ECSj}, h, (r, s)_{M2}\}, \{m, f, Q_{TDi}, (r, s)\}\cdot x]$, as depicted in Figure 4. Therefore, the communication overhead introduced by the ECS$_j$ is equal to $(4 + 32 + 64)$ bytes, i.e., 100 bytes. This represents a total communication overhead from all TD$_i$ and all ECS$_j$ equal to $(140 \times z + 100{\cdot}z)$ bytes, which is a communication overhead that is slightly smaller than the method presented in [21] and outperforms the proposals from [22, 23].

6. Conclusions

In this paper, an original method to include a dual signature into ECDSA has been proposed. The use of the presented method allows for the preservation of privacy in data transferred from IoMT devices to the cloud through edge computing servers. Specifically, collected health data remain invisible to the edge device, and the identity of the transmission medical IoT device, e.g., wearables or smartphones, is anonymous to the cloud. This solution is affordable for constrained IoMT devices, and at the same time, its hardware implementation is completely feasible because of its ECC-based approach.

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 AEI/FEDER EU project grant (AIM) (TEC2016-76465-C2-1-R).

References

[1] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[2] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.

[3] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, "Internet of medical things: a review of recent contributions dealing with cyber-physical systems in medicine," *IEEE Internet Things Journal*, vol. 5, no. 5, pp. 3810–3822, 2018.

[4] Z. Liu, Y. Cao, L. Cui, J. Song, and G. Zhao, "A benchmark database and baseline evaluation for fall detection based on wearable sensors for the internet of medical things platform," *IEEE Access*, vol. 6, pp. 51286–51296, 2018.

[5] C. Wang, Y. Qin, H. Jin et al., "A low power cardiovascular healthcare system with cross-layer optimization from sensing patch to cloud platform," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 2, pp. 314–329, 2019.

[6] E. Ahmed and M. H. Rehmani, "Mobile edge computing: opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017.

[7] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.

[8] NIST, "Digital Signature Standards (DSS)," 2009.

[9] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in IoT devices with minimal airtime consumption," *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 1–4, 2017.

[10] Y. Zhang, L. Xu, Q. Dong et al., "Recryptor: a reconfigurable cryptographic cortex-M0 processor with in-memory and

near-memory computing for IoT security," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 4, pp. 995–1005, 2018.

[11] H. D. Tiwari and J. H. Kim, "Novel method for DNA-based elliptic curve cryptography for IoT devices," *ETRI Journal*, vol. 40, no. 3, pp. 396–409, 2018.

[12] Z. Liu, J. Großschadl, Z. Hu, K. Jarvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 773–785, 2017.

[13] D. Koo and J. Hur, "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 739–752, 2018.

[14] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2019.

[15] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-empowered edge computing environment for cyber-threat detection in smart vehicles," *IEEE Network*, vol. 32, no. 3, pp. 42–51, 2018.

[16] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things of Journal*, vol. 34, pp. 1–9, 2019.

[17] R.-H. Hsu, J. Lee, T. Q. S. Quek, and J.-C. Chen, "Reconfigurable security: edge-computing-based framework for IoT," *IEEE Network*, vol. 32, no. 5, pp. 92–99, 2018.

[18] Z. Guan, Y. Zhang, L. Wu et al., "APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

[19] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 12, pp. 2681–2691, 2015.

[20] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PACRT: chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2019, 2019.

[21] F. Wu, X. Li, L. Xu, A. K. Sangaiah, and J. J. P. C. Rodrigues, "Authentication protocol for distributed cloud computing: an explanation of the security situations for internet-of-things-enabled devices," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 38–44, 2018.

[22] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, no. 2, pp. 712–719, 2018.

[23] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2428–2435, 2017.

[24] M. Alkharji, H. Liu, and M. Al Hammoshi, "A comprehensive study of fully homomorphic encryption schemes," *International Journal of Advanced Computer Technology*, vol. 10, no. 1, pp. 1–24, 2018.

[25] B. D. Deebak, F. Al-Turjman, M. Aloqaily, and O. Alfandi, "An authentic-based privacy preservation protocol for smart e-healthcare systems in IoT," *IEEE Access*, vol. 7, pp. 135632–135649, 2019.

[26] I. S. Farahat, A. S. Tolba, M. Elhoseny, and W. Eladrosy, "A secure real-time internet of medical smart things (IOMST)," *Computers & Electrical Engineering*, vol. 72, pp. 455–467, 2018.

[27] S. Pirbhulal, O. W. Samuel, W. Wu, A. K. Sangaiah, and G. Li, "A joint resource-aware and medical data security framework for wearable healthcare systems," *Future Generation Computer Systems*, vol. 95, pp. 382–391, 2019.

[28] Z. Guan, Z. Lv, X. Du, L. Wu, and M. Guizani, "Achieving data utility-privacy tradeoff in Internet of Medical Things: a machine learning approach," *Future Generation Computer Systems*, vol. 98, pp. 60–68, 2019.

[29] F. Alsubaei, A. Abuhussein, V. Shandilya, and S. Shiva, "IoMT-SAF: Internet of Medical Things Security Assessment Framework," *Internet of Things*, vol. 2019, 2019.

[30] A. Limaye and T. Adegbija, "HERMIT: a benchmark suite for the internet of medical things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4212–4222, 2018.

[31] J. Han, Y. Li, J. Liu, and M. Zhao, "An efficient lucas sequence-based batch Auditing scheme for the internet of medical things," *IEEE Access*, vol. 7, pp. 10077–10092, 2018.

[32] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing Bloom filter: challenges, solutions, and comparisons," *IEEE Communications Surveys and Tutorials*, vol. 18, 2018.

[33] B. Arazi, "Implementation of digital signatures," *Electronics Letters*, vol. 18, no. 21, p. 900, 1982.

[34] VISA and Mastercard, "SET: Secure Electronic Transaction (TM), Version 1.0, Book 1: Business Description, Book 2: Programmer's Guide, Book 3: Formal Protocol Definition," 2002.

[35] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication protocols for internet of things: a comprehensive survey," *Security and Communication Networks*, vol. 2017, p. 41, 2017.

[36] National Institute of Standards and Technology (NIST), "FIPS 186-4 Digital Signature Standard (DSS)," 2013.

[37] J. R. Shaikh, M. Nenova, G. Iliev, and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained E-commerce applications," in *Proceedings of the IEEE International Conference On Microwaves, Antennas, Communications And Electronic Systems (COMCAS)*, pp. 1–4, New York, NY, USA, 2017.

[38] T. Pornin, "BearSS-on Performance," 2018.

[39] A. Sghaier, M. Zeghid, C. Massoud, and M. Machout, "Design and implementation of low area/power elliptic curve digital signature hardware core," *Electron MDPI*, vol. 6, no. 46, pp. 1–23, 2017.

[40] Crypto Tool, "Crypto++ 5.6.0 Benchmarks," 2019.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8387672, 9 pages
https://doi.org/10.1155/2020/8387672

Research Article

User Audit Model Based on Attribute Measurement and Similarity Measurement

Xiaohui Yang and Ying Sun

School of Cyber Security and Computer, Hebei University, Baoding 071002, China

Correspondence should be addressed to Xiaohui Yang; yxh@hbu.edu.cn

Received 25 October 2019; Accepted 18 January 2020; Published 9 March 2020

Guest Editor: Geethapriya Thamilarasu

The Internet of Things (IoT) is an open network. And, there are a large number of malicious nodes in the network. These malicious nodes may tamper with the correct data and pass them to other nodes. The normal nodes will use the wrong data for information dissemination due to a lack of ability to verify the correctness of the messages received, resulting in the dissemination of false information on medical, social, and other networks. Auditing user attributes and behavior information to identify malicious user nodes is an important way to secure networks. In response to the user nodes audit problem, a user audit model based on attribute measurement and similarity measurement (AM-SM-UAM) is proposed. Firstly, the user attribute measurement algorithm is constructed, using a hierarchical decision model to construct a judgment matrix to analyze user attribute data. Secondly, the blog similarity measurement algorithm is constructed, evaluating the similarity of blog posts published by different users based on the improved Levenshtein distance. Finally, a user audit model based on a security degree is built, and malicious users are defined by security thresholds. Experimental results show that this model can comprehensively analyze the attribute and behavior data of users and have more accurate and stable performance in the practical application of the network platforms.

1. Introduction

The Internet of Things (IoT) is the latest evolution of the Internet, including a great deal of connected physical devices and applications [1]. IoT allows object collection and data exchange, etc. [2], which can perform medical data management, medical information monitoring, and user information analysis. At present, problems such as violating the privacy of medical data and publishing false medical advertisements often appear in the network, and malicious users become more and more complicated and hidden, which brings great security threats to networks. Accurate and rapid identification of malicious users not only benefits the security of the user's data and information but also facilitates timely response to threats in networks.

When objects connected to the Internet of Things continue to generate information and report to Internet users, a noteworthy development is that they will also join traditional social networks and interact with "people" in social networks. Social networks are not just person-to-person social, but person-to-person, person-to-thing, and thing-to-thing. Therefore, malicious users in social networks will inevitably pose a threat to the security of the Internet of Things.

To identify malicious users in social networks and ensure the security of the Internet of Things, a user audit model based on attribute and similarity measures is proposed. The model measures the similarity between complex user attributes and users, analyzes the user's attribute information and behavior information, determines the user's security index, and finds the similarity of self-issued behavior among users, which improves the accuracy of the model to identify malicious users. At the same time, the concept of user security is proposed to measure user security in the Internet of Things, which is an important indicator to identify malicious user nodes.

The contributions of this paper are listed as follows:

(1) Construct user attribute measurement algorithm, obtain user attribute data, calculate attribute weight vector by hierarchical weight decision model, and analyze attribute information.

(2) Construct similarity measurement algorithm, consider user blog text information, use word segmentation technology, extract original blog content keywords, and improve Levenshtein distance. By studying the contents of blog posts, it reflects the preferences and characteristics of users' spontaneous behaviors.

(3) Propose the concept of user security degree as an important distinguishing indicator between normal users and malicious users. At the same time, the security threshold is defined, security threshold judgment based on user security degree to identify malicious users.

(4) Analyze the performance of the model in the real microblog dataset and compare it with other algorithm models. AM-SM-UAM has better performance in improving the accuracy, stability, and model parameter tuning of malicious user nodes.

The rest of this paper is organized as follows. In Section 2, we provide a brief introduction to existing related work. The model is described in section 3. In Section 4, we introduce the draft model AM-SM-UAM in detail. In Section 5, we introduce the experimental results. Finally, we conclude our work in Section 6.

2. Related Work

In recent years, malicious user identification methods based on abnormal behavior detection have attracted considerable attention. Hajmohammadi et al. [3]used active learning to automatically obtain malicious users, which has the problems of large computational overhead, information redundancy, and information overload. Gupta et al. used feature extraction methods, such as text features [4, 5] and network structure features [6–8], to extract distinguishing features from a large number of marked normal users and malicious users to train the user classification model. Due to different evaluation criteria of the extracted distinguishing user features in diverse application backgrounds, the detection accuracy is low and the stability is poor. Lee et al. [9] attracted malicious users to actively attract attention by adding trapping nodes to the network and obtained the behavior characteristics of malicious users separate from normal users. The detection framework based on the trapping system was used to determine malicious users of MySpace and Twitter. Zhang et al. [10] and Tahir et al. [11] analyzed the effect of collaborative learning on clustering, and the accuracy of the identification of malicious users was minimal. Meng and Kwok [12] corrected the false alarm rate of abnormal intrusion detection based on SVM. Although partially labeled training samples were used to reduce the system overhead, most training samples were assumed to be uniform and average, and the actual situation is sometimes difficult to meet the condition, often overfitting phenomenon. Zhu et al. [13] proposed a social group identification method based on local attribute community detection. Owing to a large number of adjacent nodes, the computational overhead is relatively large. Abnormal behavior detection methods based on user relationship, such as Ju et al. [14], based on the calculation model of compactness centrality and credit, judged the influence of users by user relationship adjacency matrix; Li et al. [15] proposed the PageRank based on account anomaly detection algorithm, which builds a social relationship matrix based on the user relationship and ranks the account to detect malicious users through the iterative calculation of PageRank value. This method does not consider the user's attribute characteristics, and the ranking result of the user is affected by the time delay, so the accuracy rate is minimal in the IoT with an uneven scale.

In summary, existing malicious user identification methods have three important shortcomings. First, user data samples are required to be high, the test results are unstable, and the evaluation indexes such as computational efficiency and accuracy cannot be the best of both worlds. Second, feature extraction, clustering, and other methods only consider the user attribute characteristics or only consider the user relationship information, without considering the user spontaneous behavior, the detection of social user attribute information, and spontaneous behavior information. Third, only numerical characteristics are considered, and text data such as user blog information are not considered.

In the era of mobile Internet, the Internet of Things needs to store, calculate, and analyze data through the service management layer when it implements information processing functions. It uses existing or perceived information to create new information. During development, it is necessary not only to configure the device network but also to perform user system development, data processing, etc. At this time, the Internet of Things to hardware also has social attributes. Therefore, to maintain the security of the Internet of things and identify malicious users in the network, in response to the above problems, a user audit model based on attribute measurement and similarity measurement (AM-SM-UAM) is proposed by taking the social platform of microblog with a large user volume as an example. AM-SM-UAM defines the concept of user security degree and builds an attribute measurement algorithm and a similarity measurement algorithm to audit user attribute information and behavior information and to identify malicious user nodes in the microblog.

3. Model Description

The key to the construction of the AM-SM-UAM is to rationally quantify the user's attribute information and behavior information, to realize the identification of malicious users and to ensure the smooth operation of the microblog. A series of operations, such as analyzing users' information and measuring user attributes and the similarity of blog content, is meant by user audit.

Microblog user set, $U = \{u_i\}$ $(i = 1, \ldots\ldots, n)$, represents the collection of microblog users including malicious users and normal users, and malicious user um, um $\in U$, represents the malicious user identified by the user audit. Then, the problem of auditing microblog users to identify malicious users is defined as follows: how to perform user auditing on the user set U in the microblog and determine the malicious user um by constructing the attribute measurement algorithm and similarity measurement algorithm. AM-SM-UAM consists of three layers as shown in Figure 1.

(1) Data layer: read the original data and preprocess the data. The user vector is constructed, and the valid user attribute information and user blog text information in the original data are selected.

(2) Feature layer: user attribute information and blog text information are constructed based on user features. Attribute vectors are established based on user attribute features, and user attributes are represented by numerical values. The text information of user blog is analyzed by using the word segmentation technology, the keywords are extracted to represent user blog, and the user text data are processed to achieve the purpose of simultaneously processing and analyzing both numerical data and text data.

(3) Audit layer: two targeted algorithm strategies are proposed to implement user auditing. First, an attribute measurement algorithm is constructed to quantify user attribute information. Establish a hierarchical decision model, construct a judgment matrix, and calculate the user's own attribute values. Use the hierarchical decision model to calculate the user attribute weight vector, so that the relative importance of the user's various attribute information can be clearly expressed. Second, the similarity measurement algorithm is constructed to process users' blog information and evaluate the similarity of users with different attribute values in blog keywords, so as to achieve the purpose of computing the similarity of textual data. The user's attribute information and blog text information are considered comprehensively from the two aspects of user attribute and spontaneous behavior to obtain user security degree.

4. Model Construction

When AM-SM-UAM audits the attribute information and behavior information of microblog users, it comprehensively considers the user attribute features and blog content information and measures the user's security degree by measuring the user's attributes and calculating the similarity between user blogs with different attribute values.

Attribute measurement (AM) represents the user's attribute information numerically; similarity measurement (SM) represents the similarity of keywords of the original blog posts among users and reflects the characteristics of users' spontaneous behaviors. User security degree (Sec),

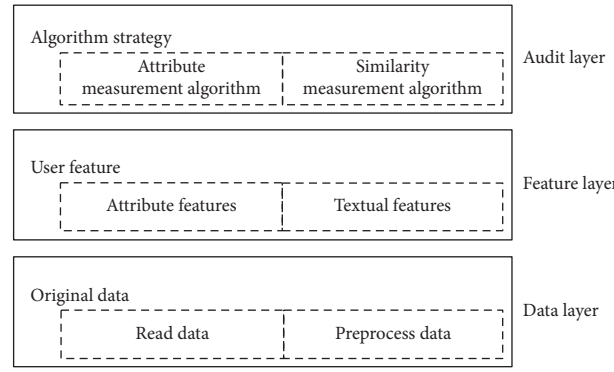


Figure 1: AM-SM-UAM framework.

which reflects the security degree of users, is calculated according to the user attribute measurement AM (u) and published content similarity measurement SM (u). The formula is shown as follows:

$$\mathrm{Sec}(u) = \mathrm{AM}(u) \cdot \mathrm{SM}(u). \tag{1}$$

4.1. Attribute Measurement. User attribute measurement is the basis for user security degree evaluation. The attributes of the measurement are shown in Table 1. In addition to the users' information integrity, other attribute information can be read in the experimental dataset, so the personal information integrity of users is defined and calculated.

Personal information integrity (A_p) refers to the proportion of the personally valid information that the user has publicly filled out, which accounts for all the information to be filled out. All the information to be filled in includes 7 items such as microblog ID, real name authentication, gender, birthday, age, region, and company.

Personal information integrity was calculated, and the construction vector E was used to represent the user's data, as shown in the following equation:

$$E = (d_1, d_2, d_3, d_4, d_5, d_6, d_7), \tag{2}$$

where d_k $(k = 1, 2, \ldots, 7)$ indicates whether item k is filled in completely, and $d_k = 0$ indicates that no valid information is filled in item k; $d_k = 1$ indicates that valid information has been filled in item k.

The user vector model was constructed. By obtaining the user's data, unmarked valid user tags were selected to judge the 7 data information, and the information was marked as valid or invalid according to the actual filling situation until all the user tags were marked. The user's information integrity is determined by calculating the scarcity of vector E, as shown in the following equation:

$$A_p(u) = \frac{1}{7} \sum_{k=1}^{7} d_k, \tag{3}$$

where A_p (u) represents the integrity of user u's personal information; 7 is the total dimension of E.

According to the relative importance of the five user attribute information of microblog level A_l, big-V

Table 1: User attribute features.

Feature symbol	Feature category	Feature name
A_l	User attribute	Microblog level
A_V	User attribute	Big-V certification
A_p	User attribute	Personal information integrity
A_f	User attribute	Number of followers
A_s	User attribute	Number of fans

certification A_V, personal information integrity A_p, number of followers A_f, and number of fans A_s, the hierarchical decision model was used to calculate the weight vector β, and the specific value is determined by experiments.

The structure of the hierarchical decision model includes the target layer, the criterion layer, and the scheme layer, as shown in Figure 2. The first layer represents the target layer of the metric user; the second layer represents the criterion layer that the five user attribute features affect the target determination, and the third layer represents the scheme layer of the user activity.

According to the attribute vectors corresponding to the five user features of microblog level A_l, big-V certification A_V, personal information integrity A_p, number of followers A_f, and number of fans A_s, and combined with the weight vector β, the user attributes are numerically represented to reflect the user's own security degree, as in the following equation:

$$\mathrm{AM}(u) = \left(A_l, A_v, A_p, A_f, A_s\right) \cdot (\beta)^T. \tag{4}$$

4.2. Similarity Measurement. Users' original blogs reflect their behavior features. Keywords in user blog content are extracted, and similarity of blog content among users with different attribute values is estimated to discover user behavior characteristics and complete user similarity measurement. The similarity of the blog can be converted into the problem of similarity between two strings, and the operation steps between strings are utilized for calculation.

Levenshtein distance refers to the minimum number of editing operations required to convert the source string into the target string between the source string and the target string [16], and the allowed to edit operation includes replacing, inserting, and deleting.

Since the user blog post appears in the form of long and short sentences, and the sequence of long and short sentences in a blog post does not influence the similarity of users, there are two disadvantages indirectly using the edit distance calculation. First, the experimental error of taking a whole blog post as a comparison string is large. Second, the number of substitutions of the sequence of long and short sentences in a blog post will be counted into the number of operations, increase the editing distance, and reduce the similarity, and has errors compared with the actual situation.

In this regard, two improvement methods of editing distance are proposed when constructing the similarity measurement algorithm. (a) Jieba [17] was used to process the user's blog content, dividing the whole post into several keywords. (b) The sequence of keywords in actual blog posts does not affect the judgment of similarity. To avoid the phenomenon of low similarity caused by inconsistent word order, the overlapping keywords in the two strings are deleted, and then the similarity measurement is carried out.

The similarity measurement algorithm steps are as follows:

Step 1: set up two sets of original keywords composed of keywords of blog contents, and name them, respectively, *keySetS* and *keySetT*, where the number of keywords is defined as the size of the set, named *keyNumS* and *keyNumT*.

Step 2: traverse the keywords in original keywords sets, get the coincidence keywords *keySame*, and delete them in sets, respectively. At the same time, record the number of coincident keywords named *SameNum*.

Step 3: record the current keyword sets *keySetX* and *keySetY* after deleting the coincident keywords, and convert the two sets into a source string *strX* and a target string *strY*. Set $x_1 \ldots x_m$ and $y_1 \ldots y_n$ representing them, respectively, where m is the length of *strX* and n is the length of *strY*.

Step 4: define $(m+1)\cdot(n+1)$ order D$[m][n]$, and save the minimum number of edit operations needed to convert *strX* to *strY*, as shown in equation (5).

Step 5: calculate the similarity SM of blog posts. The formulas are shown in equations (6) and (7).

$$D[m][n] = \begin{cases} 0, & m = 0, n = 0, \\ n, & m = 0, n = 0, \\ m, & m > 0, n = 0, \\ \min\{D[m-1][n] + 1, D[m][n-1] + 1, D[m-1][n-1] + flag\} & m > 0, n > 0, \end{cases} \tag{5}$$

where *flag* is used to mark the number of valid substitutions during the comparison of the *strX* and *strY* characters, $\text{flag} = \begin{cases} 0, X[m] = Y[n] \\ 1, X[m] \neq Y[n] \end{cases}$.

In equation (5), when $m > 0$ and $n > 0$, it corresponds to three operation modes of strings, respectively: (a) delete operation: D$[m-1][n]+1$ means to delete the last character of *strX* and add 1 to the number of editing; (b) insert

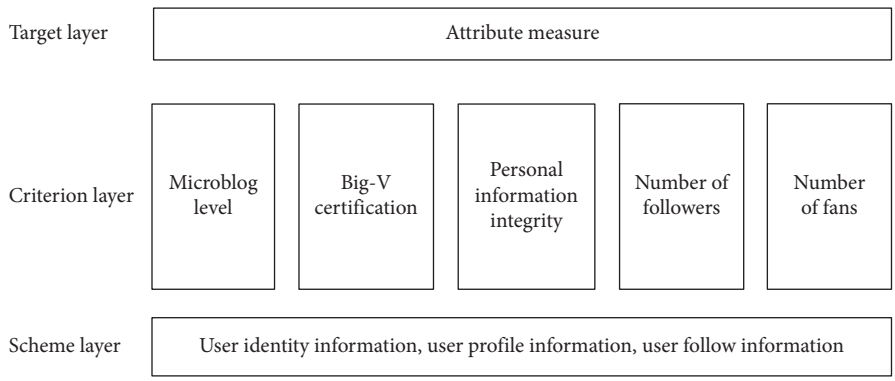


Figure 2: Hierarchical decision model.

operation: $D[m][n-1]+1$ means that the last character of *strY* is inserted into *strX*, and the number of editing is increased by one; (c) replace operation: $D[m-1][n-1]+flag$ indicates that the last character of the string *Y* is substituted to *strX*. The number of editing is determined by the *flag*, which is used to mark the number of valid substitutions:

$$\text{sim} = \left(1 - \frac{D[m][n]}{\max(m,n)}\right) + \frac{SameNum}{\max(keyNumS, keyNumT)}, \tag{6}$$

$$\text{SM} = \frac{\text{sim}}{\text{sim}_{\max}}, \tag{7}$$

where $D[m][n]$ represents the Levenshtein distance between the source string *strX* and the target string *strY*.

5. Experiments

5.1. Experimental Environment and Data. The environment used in the experiment was Intel(R) Core(TM) i5-7300HQ CPU @2.50 GHz, 8 GB of memory, the operating system is Windows 10, and Model code is based on C++ implementation.

The dataset published in [18] was used to verify the feasibility of the model. The dataset contains 1,787,443 microblog user data, and each user data includes basic information of the user (such as user ID, gender, number of followers, and number of fans) and 1000 microblogs newly released by each user. Among them, there are nearly 4 billion relationships of mutual concern among users. Due to a large amount of data in the dataset, 10 groups are randomly selected from the dataset, each group has 10,000 pieces of user data, and each piece of user data includes the basic information of the user and the newly published blog content, which is recorded as "Data1," "Data2," "Data3," "Data4," "Data5," "Data6," "Data7," "Data8," "Data9," and "Data10."

5.2. Evaluation Index. To solve the data imbalance problem, confusion matrix analysis experiment results were established [19]. In the matrix, TP stands for the number of users that are originally malicious users and are judged to be malicious users during detection; FN stands for the number of users that are originally malicious users but are judged to be normal users during detection; FP stands for the number of users that are originally normal users but are judged to be malicious users during detection; and TN stands for the number of users that are originally normal users and are judged to be normal users during detection, as shown in Table 2.

Table 2: Symbol description.

Detection result	Actual situation	
	Malicious users	Normal users
Malicious users	TP	FP
Normal users	FN	TN

To evaluate the performance of UAM, three evaluation indexes, namely, precision rate (Pre), recall rate (Rec), and harmonic mean value F1_score were selected. Among them, the precision rate and recall rate were used to evaluate the accuracy of the experiment, and the harmonic mean value was used to evaluate the comprehensive performance of the experiment, and the definitions are shown in the following equations:

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8}$$

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$\text{F1\_score} = 2 \cdot \frac{\text{Pre} \cdot \text{Rec}}{\text{Pre} + \text{Rec}}. \tag{10}$$

5.3. Parameter Settings. Parameters involved in the experiment include security threshold φ and weight vector β. The safety threshold φ was optimized through experiments, and its value was determined by referring to the performance of the model evaluated by means of harmonic mean F1_score, as shown in the following analysis; the weight vector β is determined by a hierarchical decision model, and the calculation process is as follows.

According to the hierarchical model, user attributes are measured, in which W_1, W_2, W_3, W_4, and W_5 represents A_l,

Table 3: Judgment matrix.

	W_1	W_2	W_3	W_4	W_5
W_1	1	3/5	3/7	3	3
W_2	5/3	1	5/7	5	5
W_3	7/3	7/5	1	7	7
W_4	1/3	1/5	1/7	1	1
W_5	1/3	1/5	1/7	1	1

Table 4: Treated matrix.

	W_1	W_2	W_3	W_4	W_5	*Sum*	β
W_1	1	3/5	3/7	3	3	8.029	0.163
W_2	5/3	1	5/7	5	5	13.381	0.242
W_3	7/3	7/5	1	7	7	18.733	0.463
W_4	1/3	1/5	1/7	1	1	2.676	0.066
W_5	1/3	1/5	1/7	1	1	2.676	0.066

A_V, A_s, A_f, and A_h five attribute features of users. The weights of the five features are set as $W_1 = 3$, $W_2 = 5$, $W_3 = 7$, $W_4 = 1$, and $W_5 = 1$. The proportional nine scale method [20] proposed by T.L. Saaty is used as a comparison scale to compare the relative importance of each index in the criterion layer. The structural judgment matrix is shown in Table 3.

By calculating the weight vector β of each attribute through the judgment matrix, *Sum* the matrix by row and normalize the vector Sum, as shown in Table 4.

The relative importance of the five attributes was obtained, and the weight vector β was obtained as follows: $\beta =$ (0.163, 0.242, 0.463, 0.066, 0.066).

5.4. Experimental Analysis. To compare the performance difference between AM-SM-UAM and the existing advanced model, a comparative experiment was set up. AM-SM-UAM was compared with the DBSCAN-based clustering algorithm and PageRank-based anomaly detection algorithm. Through the three algorithms corresponding to the various indicators of the experiment, the accuracy of the three algorithms to identify malicious users of a microblog is analyzed.

The clustering algorithm based on DBSCAN is an anomaly detection method based on density clustering, which can find abnormal points while clustering. The PageRank-based microblog account anomaly detection algorithm constructs a social relationship matrix according to the user relationship and ranks the account by iteratively calculating the PageRank value to detect malicious users. Both algorithms have good results in malicious user identification, so the above two algorithms are used to compare experiments with AM-SM-UAM. Using these three algorithms, 10 groups of experiments were conducted on the dataset of "Data1-Data10" in turn, which were recorded as "G1-G10". Pre, Rec, and F1_score were used as the evaluation criteria of the experiment, and the experimental results are shown in Figure 3–5.

The results show that when AM-SM-UAM identifies malicious users, the precision rate difference between the 10 groups is no more than 2%, the recall rate is no more than 3%, and the F1_score is no more than 1%. Meanwhile, the precision rate, recall rate, and F1_score are all high. DBSCAN clustering algorithm and PageRank ranking algorithm have a lower precision rate when detecting malicious users of microblog, and the recall rate fluctuates

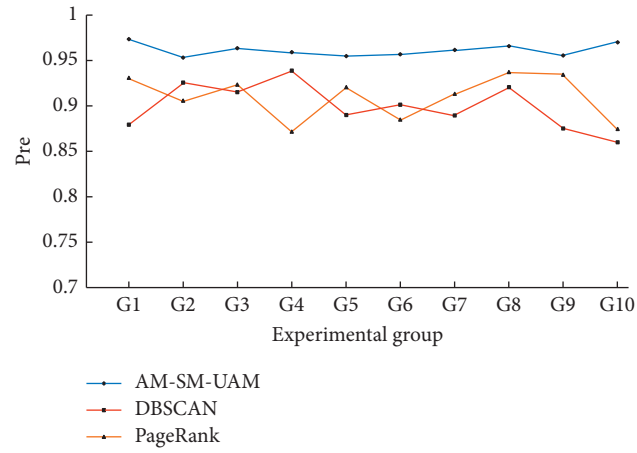


Figure 3: Precision rate.

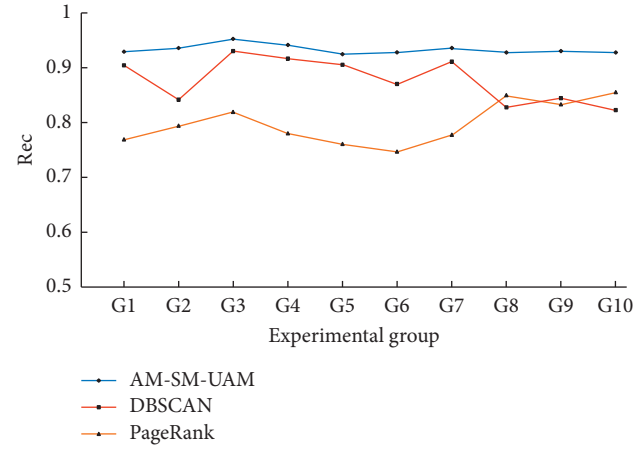


Figure 4: Recall rate.

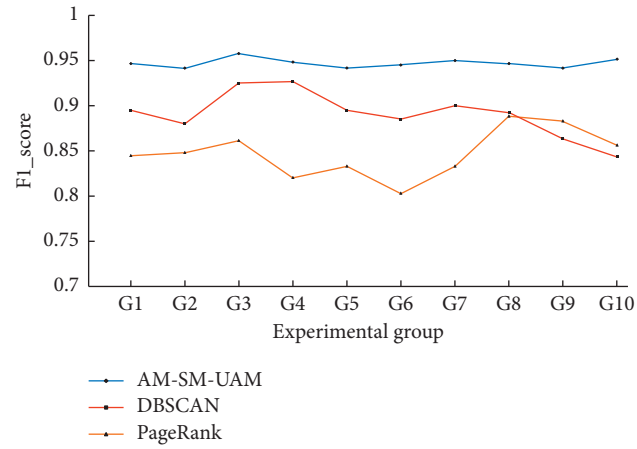


Figure 5: F1_score.

greatly, which makes the F1_score lower and unstable. According to the experimental results, the audit of users in microblog by AM-SM-UAM is completed based on the user's attribute information and the user's blog keywords. It not only considers the numerical information such as the user's attributes and reduces the influence of time delay caused by considering only the user's behavior, but also considers text information such as blog content, and the incompleteness caused by the calculation of only numeric attributes is avoided, thus improving the accuracy of identifying malicious users.

To test the stability of AM-SM-UAM audit microblog users, the average and variance of 10 groups of experimental results corresponding to the three algorithms were compared. The experimental results are shown in Figures 6 and 7.

It can be observed in Figure 6 that the 10 sets of experiments corresponding to the three algorithms are compared in terms of precision rate, recall rate, and F1_score value. Among them, the average value of the three indexes of the DBSCAN clustering algorithm is medium; the PageRank ranking algorithm although the average value is 92%, its recall rate is low, and the overall performance of the algorithm is poor. Among the 10 experiments using AM-SM-UAM, the precision rate, recall rate, and F1_score were the highest compared with the other two algorithms; the average accuracy can reach 96%.

As can be seen from Figure 7, the variance of the DBSCAN clustering algorithm and PageRank ranking algorithm on the three experimental evaluation indexes is large, indicating that the experimental results of the above two algorithms fluctuate greatly in the 10 groups of experiments, respectively, and the stability of the algorithm is poor. The variance of the 10 groups of experiments corresponding to AM-SM-UAM is small, indicating that the results of each group of experiments are less fluctuating and the stability of the algorithm is better.

According to the mean value and variance of the 10 groups of experimental results corresponding to the three algorithms, in the process of auditing microblog users' experiment, compared with the other two algorithms, AM-SM-UAM algorithm also has better stability and adaptability under the premise of ensuring a higher accuracy of identifying malicious users.

5.5. Parameter Tuning. DBSCAN clustering algorithm, PageRank ranking algorithm, and AM-SM-UAM algorithm all require parameter adjustment to achieve malicious user identification. The DBSCAN clustering algorithm needs to set two parameters, namely, neighborhood threshold (Eps) and point threshold (Minpts). According to the parameters, the region with a certain density is divided into clusters, and the clustering results are sensitive to the parameter values. The PageRank ranking algorithm calculates the user PR value by matrix iteratively to rank the user to complete the detection of the malicious user and the setting of the damping factor and the iteration termination threshold has a decisive influence on the user PR value calculation, and the ranking result is sensitive to the parameter value. The above two algorithms are greatly affected by the parameters, and the performance of the algorithm fluctuates greatly.

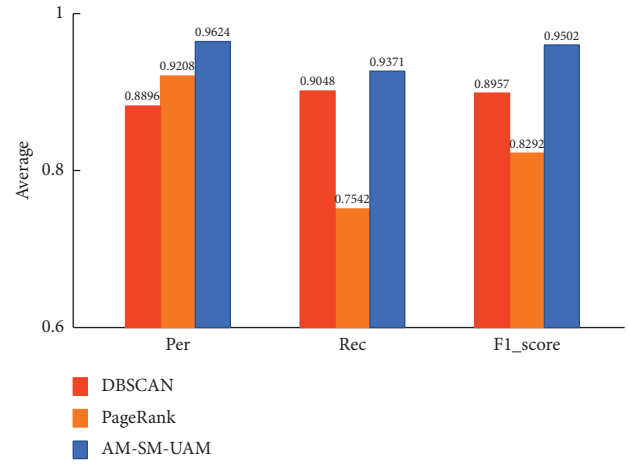


Figure 6: The average.

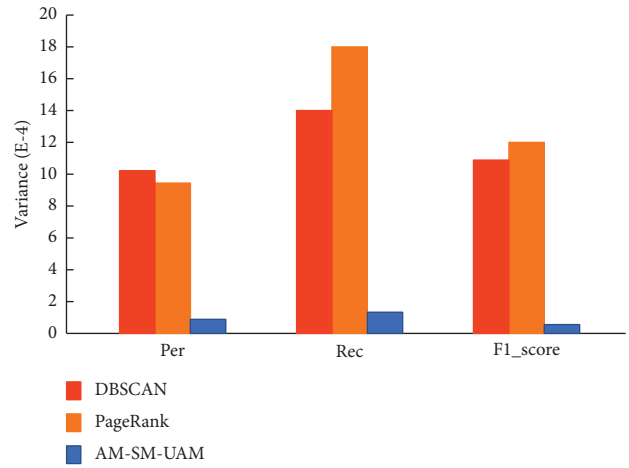


Figure 7: The variance.

The security threshold φ in AM-SM-UAM is related to the accuracy of identifying malicious users. By equation (1), the formula of the safety of users for the $\mathrm{Sec}(u_i) = \mathrm{AM}(u_i) \cdot \mathrm{SM}(u_i)$, in which $\mathrm{AM}(u_i) = (A_l, A_V, A_p, A_f, A_s) \cdot (\beta)^T$ the weight vector of beta calculated by hierarchical decision model. Therefore, on the premise that the weight vector β has been determined, the safety threshold φ should be determined by the size of F1_score and the relationship between the security threshold φ and F1_score is shown in Figure 8.

As can be observed in Figure 8, when the security threshold φ is 0.4, the F1_score value is the largest. Therefore, when the security threshold $\varphi = 0.4$, that is, the user security degree less than 0.4 users defined as malicious users, AM-SM-UAM has the best performance.

To verify the rationality of the security threshold of 0.4, 10 groups of experiments of AM-SM-UAM auditing microblog users were analyzed. Take the user security degree of normal users and malicious users in microblog calculated from the "G1-G10" 10 groups of experiments, and respectively. calculate the average of the security degree of normal users and malicious users in each group of experiments, as shown in Figure 9.

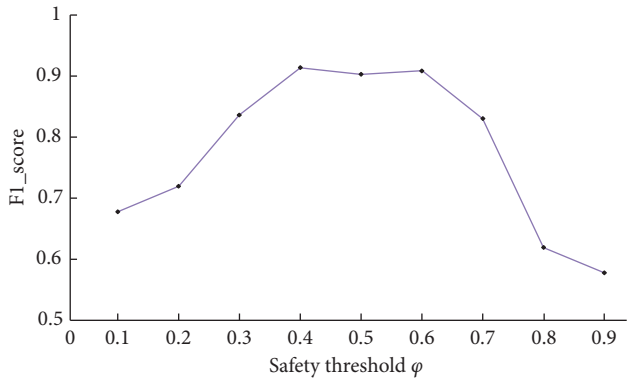


Figure 8: Relationship between φ and F1_score.

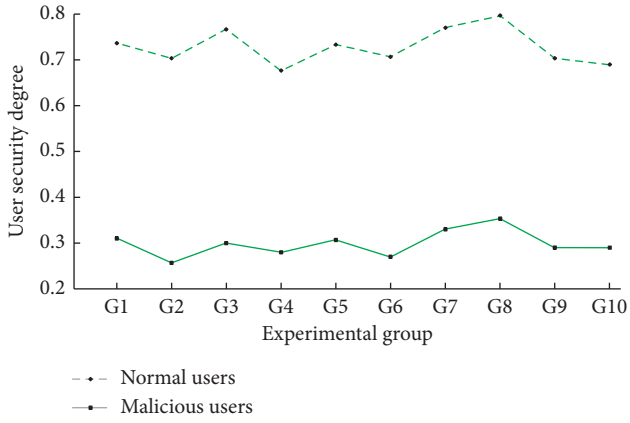


Figure 9: User security degree distribution.

The results show that the average security degree of normal users is distributed in [0.6, 0.8], while that of malicious users is distributed in [0.2, 0.4]. According to the experimental results, the average security degree between normal users and malicious users in microblog has a large gap, so the degree range of the security threshold can be [0.4, 0.6]. According to the experimental results, compared with the other two algorithms, in the process of parameter tuning, UAM is easy to find the optimal parameters, which is more conducive to identifying malicious users in the microblog.

6. Conclusion

This paper proposes a microblog user audit model based on attribute measurement and similarity measurement (AM-SM-UAM), which is used to detect a large number of malicious nodes in the IoT and identify false information on medical and social networks. Firstly, the concept of user security degree was proposed to reflect the security level of microblog users, as the standard of differentiation between malicious users and normal users. Secondly, the user attribute measurement algorithm was constructed, using a hierarchical decision model to construct a judgment matrix to analyze user attribute data. Finally, the similarity measurement algorithm was constructed, keywords of user original blog with word segmentation technology were extracted, Levenshtein distance was improved, user blog content similarity was calculated, and user behavior information data were analyzed. Through the measurement of the user attribute information and the calculation of the similarity of the blog keywords, the user security degree was obtained, and the malicious user um was determined. Experiments showed that AM-SM-UAM achieved more accurate and stable performance.

In the future, the behavior of malicious user nodes in the IoT will be specifically analyzed to determine the correlation behavior between malicious users. At the same time, the probability of associative behaviors between malicious nodes in medical IoT is considered by increasing inference calculation, and the identification of malicious nodes and false behaviors in medical IoT is further discussed.

Data Availability

The data came from an article [18] by Zhang Jing of Tsinghua University, in which crawlers were used to construct a dataset of microblog users. The microblogging network they used in this study was crawled from Sina Weibo.com, which, similar to Twitter, allows users to follow each other. Particularly, when user A follows B, B's activities such as (tweet and retweet) will be visible to A. A can then choose to retweet a microblog that was tweeted (or retweeted) by B. User A is also called the follower of B and B is called the followee of A. After crawling the network structure, for each one in the 1,787,443 core users, the crawler collected her 1,000 most recent microblogs. At the end of the crawling, they produced in total 4 billion following relationships among them, with an average of 200 followers per user.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant no. 2017YFB0802300.

References

[1] K. Fan, S. Sun, Z. Yan, Q. Pan, H. Li, and Y. Yang, "A blockchain-based clock synchronization scheme in IoT," *Future Generation Computer Systems*, vol. 101, pp. 524–533, 2019.

[2] K. Fan, W. Jiang, L. Qi, H. Li, and Y. Yang, "Cloud-based RFID mutual authentication scheme for efficient privacy preserving in IoV," *Journal of the Franklin Institute*, vol. 9, no. 35, 2019.

[3] M. S. Hajmohammadi, R. Ibrahim, A. Selamat, and H. Fujita, "Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples," *Information Sciences*, vol. 317, pp. 67–77, 2015.

[4] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "TweetCred: real-time credibility assessment of content on twitter," in *Proceedings of the 6th International Conference on*

Social Informatics, pp. 228–243, Springer, Barcelona, Spain, 2014.

[5] A. A. Amleshwaram, N. Reddy, S. Yadav, G. F. Gu, and C. Yang, "CATS: characterizing automation of Twitter spammers," in *Proceedings of the 5th International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–10, IEEE, Bangalore, India, 2013.

[6] X. Hu, J. L. Tang, and H. Liu, "Online social spammer detection," in *Proceedings of the 28th Conference on Artificial Intelligence*, AAAI, Quebec, Canada, pp. 59–65, July 2014.

[7] X. Hu, J. L. Tang, Y. C. Zhang, and H. Liu, "Social spammer detection in microblogging," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 2633–2639, AAAI, Beijing, China, 2013.

[8] S. Ravikumar, K. Talamadupula, R. Balakrishnan, and S. Kambhampati, "RAProp: ranking tweets by exploiting the tweet/user/web ecosystem and inter-tweet agreement," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pp. 2345–2350, ACM, San Francisco, CA, USA, 2013.

[9] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: social honeypots+machine learning," in *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 435–442, ACM, Geneva, Switzerland, 2010.

[10] J. Zhang, Y. Yang, H. Wang et al., "Semi-supervised clustering ensemble based on collaborative training," in *Proceedings of the International Conference on Rough Sets and Knowledge Technology*, Springer-Verlag, Berlin, Germany, pp. 450–455, 2012.

[11] N. Tahir, A. Hassan, M. Asif et al., "MCD: mutually connected community detection using clustering coefficient approach in social networks," in *in Proceeding of the 2nd International Conference on Communication, Computing and Digital Systems (C-CODE)*, IEEE, Islamabad, Pakistan, March 2019.

[12] Y. Meng and L. F. Kwok, "Intrusion detection using disagreement-based semi-supervised learning: detection enhancement and false alarm reduction," in *Proceedings of the International Conference on Cyberspace Safety and Security*, Springer-Verlag, Melbourne, Australia, pp. 483–497, 2012.

[13] J. Zhu, Y. Li, and R. Liu, "Social network group identification based on local attribute community detection," in *Proceedings of the IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Chengdu, China, March 2019.

[14] C. Ju, K. Zhao, and F. Bao, "Influence intensity calculation model of social network users integrating closeness centrality and credit," *Journal of Intelligence*, vol. 38, no. 2, pp. 170–177, 2019.

[15] S. Li, X. Li, H. Yang et al., "A zombie account detection method in microblog based on the pagerank," in *in Proceedings of the 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, IEEE, Prague, Czech Republic, July 2017.

[16] X.-M. Lin and W. Wang, "Set and string similarity queries: a survey," *Chinese Journal of Computers*, vol. 34, no. 10, pp. 1853–1862, 2011.

[17] https://pypi.org/project/jieba/.

[18] J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing, "Who influenced you? predicting retweet via social influence locality," *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 3, pp. 1–26, 2015.

[19] M. Yang, J. M. Yin, and G. L. Ji, "Classification methods on imbalanced data: a survey," *Journal of Nanjing Normal University (Engineering and Technology Edition)*, vol. 8, no. 4, pp. 7–12, 2008.

[20] Z. Q. Luo and S. L. Yang, "Comparative study on several scales in AHP," *Systems Engineering-Theory & Practice*, vol. 9, pp. 51–60, 2004.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 7542726, 10 pages
https://doi.org/10.1155/2020/7542726

Research Article

Secure Information Transmissions in Wireless-Powered Cognitive Radio Networks for Internet of Medical Things

Kun Tang,[1,2] **Wenjuan Tang**,[3] **Entao Luo**,[2] **Zhiyuan Tan**,[4] **Weizhi Meng**,[5] **and Lianyong Qi**[6]

[1]*Guangdong Provincial Key Laboratory of Millimeter-Wave and Terahertz, School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China*
[2]*School of Electronics and Information Engineering, Hunan University of Science and Engineering, Yongzhou 425000, China*
[3]*College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China*
[4]*School of Computing, Edinburgh Napier University, Edinburgh EH11 4BN, UK*
[5]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Lyngby 2800 Kgs., Denmark*
[6]*School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China*

Correspondence should be addressed to Entao Luo; luoentao_huse@163.com

Received 25 September 2019; Accepted 20 December 2019; Published 24 February 2020

Guest Editor: Kuan Zhang

In this paper, we consider the issue of the secure transmissions for the cognitive radio-based Internet of Medical Things (IoMT) with wireless energy harvesting. In these systems, a primary transmitter (PT) will transmit its sensitive medical information to a primary receiver (PR) by a multi-antenna-based secondary transmitter (ST), where we consider that a potential eavesdropper may listen to the PT's sensitive information. Meanwhile, the ST also transmits its own information concurrently by utilizing spectrum sharing. We aim to propose a novel scheme for jointly designing the optimal parameters, i.e., energy harvesting (EH) time ratio and secure beamforming vectors, for maximizing the primary secrecy transmission rate while guaranteeing secondary transmission requirement. For solving the nonconvex optimization problem, we transfer the problem into convex optimization form by adopting the semidefinite relaxation (SDR) method and Charnes–Cooper transformation technique. Then, the optimal secure beamforming vectors and energy harvesting duration can be obtained easily by utilizing the CVX tools. According to the simulation results of secrecy transmission rate, i.e., secrecy capacity, we can observe that the proposed protocol for the considered system model can effectively promote the primary secrecy transmission rate when compared with traditional zero-forcing (ZF) scheme, while ensuring the transmission rate of the secondary system.

1. Introduction

With the rapid development of wireless communication and networking technologies, an increasing number of devices need to be connected globally and communicate automatically. Therefore, the emerging of the Internet of Things (IoT) as a promising paradigm can achieve a fusing of the various technologies in 5G communication systems, which have been widely applied in smart cities, agriculture, and environment monitoring [1–6]. Moreover, the medical care and health care are becoming one of the most popular applications based on the IoT [7, 8], named the Internet of Medical Things (IoMT), which can collect the data from the medical devices and applications to improve the treatment effect, disease diagnosis, and patient experience, while reducing misdiagnosis rate and treatment cost. According to the investigation of relevant organizations, the market share of IoMT will reach to roughly 117 billion dollars by the end of 2020 [9]. However, with the increasing use of IoMT equipment, the huge demand for radio spectrum has become a serious problem. In addition, the allocated radio spectrums are often underutilized due to the inflexible spectrum policies [10]. In order to facilitate an effective utilization of spectrum resources, cognitive radio technology was introduced in which unlicensed nodes could communicate with each other in an opportunistic manner over a licensed

frequency band without interrupting the primary transmissions [11–13].

Yet, power supply is another key constraint on the development of IoMT. In general, an IoMT system usually contains a large number of small-size devices that are battery-powered and difficult to be replaced. In order to solve this problem, wireless-powered technology has been paid high attention. The devices with EH capabilities can convert energy from the surrounding environment into electricity for data transmission, such as solar, wind, or RF signals [14]. Especially with the synchronous development of antenna and circuit designs, wireless EH based on RF signals has attracted more attention due to its advantages of wireless, low cost, and small form implementation [15–17]. Furthermore, the amount of harvested energy is in milliwatts, which is enough to power small-size IoMT devices, such as medical data sensors for short-distance transmissions. Therefore, the combination of cognitive radio and EH in medical wireless sensor networks can greatly improve both the spectrum and energy efficiencies.

Although adopting cognitive radio technology with EH can effectively improve the transfer efficiency for IoMT, the variety of medical devices in healthcare fields will introduce several security problems [18]. Since the energy-constraint sensors need to perform energy harvesting and then forward the sensitive patient data wirelessly, the other illegal sensors may be the potential eavesdropper to listen such confidential messages [19]. As an emerging field, a large number of healthcare manufacturers are rushing to utilize the IoT solutions in some applications without considering security. As a result, they will bring new security problems related to confidentiality, integrity, and availability. Furthermore, due to the limited capabilities, such as lack of effective computation and sufficient power supply, many sensors in IoMT cannot embed the encryption algorithm. Therefore, this lack of strong encryption across medical sensors makes themselves to be discovered and exploited by malicious users easily.

1.1. Related Work. To take the full advantage of the potential gains for wireless EH, the researchers developed simultaneous wireless information and power transmission (SWIPT) schemes in wireless networks that utilize RF signals to transmit energy and information to receivers. Chen et al. [20] applied the SWPIT in relay interference channels for multiple source-destination pair communication system, where each pair of link has a dedicated EH relay serving for relaying transmission. On this basis, the optimal power allocation ratio for each relay was deduced by adopting the distributed power allocation framework of game theory. A SWIPT scheme for amplify-and-forward (AF) bidirectional relaying network based on OFDM was proposed in [21], where a wireless-powered relay performed information processing and EH by utilizing two disjointed subcarrier groups, respectively. Based on the decode-and-forward (DF) mode, Shi et al. [22] designed an optimal resource allocation strategy to maximize the energy efficiency with the nonlinear SWIPT model under a two-way relay network. For cognitive radio networks with energy harvesting in IoT systems, Zhang et al. [23] analyzed the outage probability of a random underlay cognitive network with EH-based assistant relay. The two main challenges for cognitive radio sensor networks in IoT systems were considered in [24], where the authors developed an architecture and proposed an energy management strategy for achieving balance between the transmission performance of the networks and operational life. In [25], the insecure characteristic of electronic medical records based on eHealth systems was considered, and then a corresponding secure encrypted scheme to ensure the data security was proposed. In [26], Gurjar et al. investigated an overlaid spectrum sharing network with SWIPT for IoT systems, where a pair of SWIPT-based devices is used as the relay to assist the transmission of the primary signals. Considering information security in cognitive radio-based IoT systems, Salameh et al. [27] presented a novel algorithm for channel allocation with time-sensitive data under the scenario of jamming attacks. A secure relay selection scheme based on channel state information and battery state information was proposed for energy harvesting-based cognitive radio networks in IoT networks [28].

1.2. Motivation and Contributions. Unlike the above-mentioned literates [27, 28], we consider an actual application scenarios for sanatorium or hospital under the cognitive radio-based IoTM networks to protect the patients' sensitive medical information. Consider an indoor environment for sanatorium or hospital, where the PT intends to transmit its sensitive medical data to the PR, while the ST performs data monitoring and transfer to the SR. In this scenario, the node ST has lack of energy supply and need to scavenge energy from the primary transmitter, while ST can be regarded as the relay to opportunistically access the licensed primary channel. Meanwhile, we assume that an attacker is located near the PR to eavesdrop the PT's medical data. Thus, to enable the secure transmission of the PT's signal, we investigate a typical cognitive radio network with wireless-powered relay (CRN-WPR) and jointly design the optimal EH time ratio and secure beamforming vectors to maximize the secrecy transmission rate of the primary system, while effectively guaranteeing the secondary transmission rate. The main contributions are summarized as follows:

(i) We propose a corresponding protocol for EH and secrecy information transmission for a cognitive radio-based IoMT system, where the relay node ST is equipped with multiple antennas to perform EH at first and then transfer the sensitively primary signal with DF processing to the destination in security with its own signal.

(ii) In order to protect the sensitive medical data being sent from the PT, we formulate the optimization problem based on maximizing the secrecy transmission rate of the primary system while ensuring the transmission requirement of the secondary system. We adopt SDR and Charnes–Cooper

transformation to transform the nonconvex optimization problem into a convex optimization problem to find a solution for the optimization problem. A corresponding algorithm is then developed. In addition, the zero-forcing (ZF) scheme is also applied to solve the optimization problem as a benchmark.

(iii) The numerical results of the influence for the secrecy transmission rate on the primary system under different system parameters are given, such as primary transmission power, number of antennas, and transmission distance. The results demonstrate excellent secure transmission performance with proposed scheme than ZF scheme.

The rest of this paper is organized as follows. The Section 2 introduces the system model and transmission protocol. Section 3 formulates the optimization problem and proposes the corresponding solution with secure beamforming. Furthermore, the ZF scheme is also adopted to solve the optimization as a benchmark. The Section 4 presents the simulation results and corresponding analyses. The Section 5 summarizes this paper.

Notations: Throughout this paper, let $(\cdot)^H$ denote the conjugate transpose. $\mathbf{I}$ presents the identity matrix with appropriate dimension. $[x]^+$ represents the maximum value between x and 0, while x^* denotes the optimal value of x. $\prod_x^\perp$ denotes the orthogonal projection onto the orthogonal complement of the column space of x. $\|\cdot\|$ denotes the Euclidean norm of a vector or a matrix and $|\cdot|$ denotes the magnitude of a channel or the absolute value of a complex number. Table 1 lists the fundamental notations and parameters.

2. System Model and Transmission Protocol

2.1. System Model. We consider a cognitive radio network with wireless-powered relay (CRN-WPR) as shown in Figure 1. The primary system is composed of a primary transmitter (PT) and a primary receiver (PR), while the secondary system consists of a secondary transmitter (ST) and a secondary receiver (SR). There also exists an eavesdropper (ME) whose purpose is to intercept the PT's confidential data in the range of the primary system, where PT intends to send confidential data to PR. The primary system may be regard as the uplink of the transmission system with poor channel quality or lower rate. Therefore, the ST is willing to act as the relay for assisting the primary transmission while delivering its own data. We assume that the PT has a fixed power supply, while the ST may have limited battery storage, so it needs to obtain energy from the received RF signal. The ST is equipped with N antennas and other nodes operate in the half-duplex mode with a single antenna.

All channels undergo the flat block Rayleigh fading channel, which is characterized by quasistatic state of the channel in one transmission-slot and independent change in different transmission-slots. Let $\mathbf{h}_{\mathrm{PST}}$, $\mathbf{h}_{\mathrm{SS}}$, $\mathbf{h}_{\mathrm{SME}}$, and $\mathbf{h}_{\mathrm{SPR}}$ be the $N \times 1$ complex channel vectors of the PT-ST, ST-SR, ST-ME, and ST-PR, respectively. The channel coefficients of the PT-PR and the PT-ME links are denoted by h_{PP} and h_{PME}. The global channel state information is available for the system, which is a common assumption in physical-layer security literatures [29, 30].

2.2. Energy Harvesting and Information Transmission. As depicted in Figure 1, the EH and information transmission in one transmission-slot include three phases. In the first phase, the PT uses a portion of time $\alpha[\alpha \in (0,1)]$ of the total block time T to transmit the dedicated energy signal x_e to ST for EH. Thus, the received signal at the ST can be expressed as

$$y_{\mathrm{ST}}^{\mathrm{I}} = \sqrt{P_P}\mathbf{h}_{\mathrm{PST}}x_e + \mathbf{n}_{\mathrm{ST}}, \tag{1}$$

where P_P represents the transmission power of the node PT, x_e denotes the unit-power energy signal, and $\mathbf{n}_{\mathrm{ST}} \sim \mathcal{CN}(0, \delta_{\mathrm{ST}}\mathbf{I})$ is the received additive Gaussian white noise (AWGN) with variance of δ_{ST}. For definiteness and without loss of generality, we assume $T = 1$. Thus, the amount of harvested energy at the ST can be calculated as

$$E_{\mathrm{ST}} = \alpha\eta P_P \left\|\mathbf{h}_{\mathrm{PST}}\right\|^2, \tag{2}$$

where $\eta \in [0,1]$ is energy conversion efficiency. Note that the amount of scavenged energy from noise is neglected because the harvested energy from the thermal noise can be negligible compared to the energy signal.

At the second phase of duration $(1-\alpha)T/2$, the PT transmits confidential signal x_P with power P_P, the received signal at the ST is thus given as

$$y_{\mathrm{ST}}^{\mathrm{II}} = \sqrt{P_P}\mathbf{h}_{\mathrm{PST}}x_P + \mathbf{n}_{\mathrm{ST}}. \tag{3}$$

The achievable rate R_{ST} can be derived as

$$R_{\mathrm{ST}} = \frac{(1-\alpha)T}{2}\log_2\left(1 + \frac{P_P\left\|\mathbf{h}_{\mathrm{PST}}\right\|^2}{\delta_{\mathrm{ST}}}\right). \tag{4}$$

Due to the nature of the information broadcast, the PR and eavesdropper ME can also receive the signal x_P and the received signals at the PR and ME are given as

$$\begin{aligned} y_{\mathrm{PR}}^{\mathrm{II}} &= \sqrt{P_P}h_{\mathrm{PP}}x_P + n_{\mathrm{PR}}, \\ y_{\mathrm{ME}}^{\mathrm{II}} &= \sqrt{P_P}h_{\mathrm{PME}}x_P + n_{\mathrm{ME}}, \end{aligned} \tag{5}$$

respectively, where $n_{\mathrm{PR}} \sim \mathcal{CN}(0, \delta_{\mathrm{PR}})$ and $n_{\mathrm{ME}} \sim \mathcal{CN}(0, \delta_{\mathrm{ME}})$ denote AWGN at PR and ME, respectively.

During the third phase $(1-\alpha)/2$, the node ST first decodes the received primary confidential signal $\widehat{x}_P$ based on DF processing and then simultaneously forwards $\widehat{x}_P$ and its own signal x_S by utilizing the beamforming vectors $\mathbf{v}_P \in \mathbb{C}^{N\times 1}$ and $\mathbf{v}_S \in \mathbb{C}^{N\times 1}$, respectively. Therefore, the corresponding received signal at the PR and eavesdropper ME are expressed as

$$\begin{aligned} y_{\mathrm{PR}}^{\mathrm{III}} &= \mathbf{h}_{\mathrm{SPR}}^H\mathbf{v}_P\widehat{x}_P + \mathbf{h}_{\mathrm{SPR}}^H\mathbf{v}_S x_S + \mathbf{n}_{\mathrm{PR}}, \\ y_{\mathrm{ME}}^{\mathrm{III}} &= \mathbf{h}_{\mathrm{SME}}^H\mathbf{v}_P\widehat{x}_P + \mathbf{h}_{\mathrm{SME}}^H\mathbf{v}_S x_S + \mathbf{n}_{\mathrm{PR}}, \end{aligned} \tag{6}$$

Table 1: List of parameters and their physical meaning/expression.

Parameter	Meaning/expression
$\mathbf{h}_{PST}$, $\mathbf{h}_{SS}$, $\mathbf{h}_{SME}$, $\mathbf{h}_{SPR}$	$N \times 1$ complex channel vectors of the PT-ST, ST-SR, ST-ME, and ST-PR, respectively
h_{PP}, h_{PME}	Channel coefficients of the PT-PR and the PT-ME
α	Duration of energy harvesting
T	Total block time
x_e, x_P	Transmit dedicated energy signal and confidential signal at PT
$\widehat{x}_P$, x_S	Decoded primary signal and secondary signal at ST
P_P	PT's transmission power
$\mathbf{n}_{ST}$, n_{PR}, n_{ME}, n_{SR}	Received AWGN at ST, PR, ME, and SR
η	Energy conversion efficiency from signal power to circuit power
R_{ST}, R_{PR}, R_{ME}, R_{SR}	Achievable rate at ST, PR, ME, and SR, respectively
$\widetilde{R}_{PR}$, $\widetilde{R}_{ME}$	Overall transmission rates at PR and ME
R_{SEC}	Secrecy rate of the primary system
$\mathbf{v}_P$, $\mathbf{v}_S$	Relaying beamforming vector and cognitive beamforming vector
E_{ST0}	Initial power at the ST
r_S	Minimal transmission rate requirement for the secondary system
Γ	An auxiliary optimization variable to bound the achievable rate of the eavesdropper ME
β	Power allocation coefficient

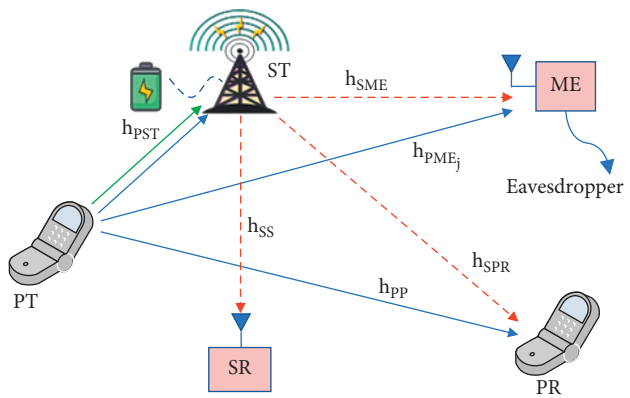


Figure 1: The system model of a CRN-WPR. The green line denotes the first phase for energy harvesting and the blue lines and red lines represent the second and third information transmission phases from the PT and ST, respectively.

respectively. The PR attempts to retrieve $\widehat{x}_P$ from y_{PR}^{III} in the presence of the secondary signal x_S. In the meanwhile, the eavesdropper also intends to intercept signal $\widehat{x}_P$. Thus, the achievable rates at the PR and ME in last two phases can be expressed as

$$R_{PR} = \frac{(1-\alpha)T}{2}\log_2\left(1 + \frac{P_P|h_{PP}|^2}{\delta_{PR}} + \frac{\left|\mathbf{h}_{SPR}^H\mathbf{v}_P\right|^2}{\left|\mathbf{h}_{SPR}^H\mathbf{v}_S\right|^2 + \delta_{PR}}\right),$$
$$R_{ME} = \frac{(1-\alpha)T}{2}\log_2\left(1 + \frac{P_P|h_{PME}|^2}{\delta_{ME}} + \frac{\left|\mathbf{h}_{SME}^H\mathbf{v}_P\right|^2}{\left|\mathbf{h}_{SME}^H\mathbf{v}_S\right|^2 + \delta_{ME}}\right). \tag{7}$$

At the SR, the received signal is given by

$$y_{SR} = \mathbf{h}_{SS}^H\mathbf{v}_S x_S + \mathbf{h}_{SS}^H\mathbf{v}_P\widehat{x}_P + \mathbf{n}_{SR}. \tag{8}$$

Similar to the PR, the SR treats $\widehat{x}_P$ as interference and then detects the desired secondary signal x_S. The achievable rate at the SR is given by

$$R_{SR} = \frac{(1-\alpha)T}{2}\log_2\left(1 + \frac{\left|\mathbf{h}_{SS}^H\mathbf{v}_S\right|^2}{\left|\mathbf{h}_{SS}^H\mathbf{v}_P\right|^2 + \delta_{SR}}\right). \tag{9}$$

3. Problem Formulation and Secure Beamforming

In this section, we first define the secrecy rate of the primary system, which is a critical performance index to illustrate the transmission security of the sensitive data [31, 32] and then formulate the optimization problem with maximizing the primary secrecy rate aiming to satisfy the minimum achievable rate for the secondary system and power constraint of the relay node ST. In order to effectively obtain the optimal parameters to keep data in safety, we also propose a mathematically efficient optimization scheme to solve the problem with a two-stage procedure.

3.1. Problem Formulation. Based on the DF cooperative communication scheme, the overall transmission rates at PR and ME equals the minimum rate of the two-hop transmissions, respectively [32], i.e.,

$$\widetilde{R}_{PR} = \min\{R_{ST}, R_{PR}\},$$
$$\widetilde{R}_{ME} = \min\{R_{ST}, R_{ME}\}. \tag{10}$$

Based on the definition of [33], the secrecy rate of the primary system for the considered secrecy CRN-WPR can be expressed as

$$R_{SEC} = \left[\widetilde{R}_{PR} - \widetilde{R}_{ME}\right]^+. \tag{11}$$

Substituting the results of equation (8) into equation (9), the overall primary secrecy rate is then given as

$$R_{SEC} = \left[\min\left(R_{ST}, R_{PR}\right) - R_{ME}\right]^+. \tag{12}$$

In the following, the EH ratio and secure beamforming vectors are jointly designed by maximizing the primary secrecy rate subject to the minimum achievable rate for the

SR and power constraint of the ST. Mathematically, the considered optimization problem can be represent as P1:

$$\begin{aligned}
&\max_{\alpha,\mathbf{v}_P,\mathbf{v}_S} \quad \left[\min\left(R_{\mathrm{ST}}, R_{\mathrm{PR}}\right) - R_{\mathrm{ME}}\right]^+ \\
&\text{s.t.} \quad \text{C1: } R_{\mathrm{SR}} \geq r_S \\
&\qquad \text{C2: } \left\|\mathbf{v}_P\right\|^2 + \left\|\mathbf{v}_S\right\|^2 \leq \frac{2\left(\alpha\eta P_P \left\|\mathbf{h}_{\mathrm{PST}}\right\|^2 + E_{\mathrm{ST0}}\right)}{1-\alpha} \\
&\qquad \text{C3: } 0 < \alpha < 1,
\end{aligned} \tag{13}$$

where C1 means that the achievable rate of SR should be larger than or equal to minimum rate r_S and C2 denotes the transmission power constraint at the ST with E_{ST0} representing the initial power at the ST.

3.2. Optimal Secure Beamforming Design. According to the analysis of formula (13), we can observe that (P1) is a nonconvex function, which is difficult to derive three optimal variables $(\alpha, \mathbf{v}_P, \mathbf{v}_S)$ concurrently. This section proposes a mathematically efficient optimization scheme with two-stage procedure for solving the (P1) as follows:

(i) In the stage I, we obtain the optimal secure beamforming $(\mathbf{v}_P^*, \mathbf{v}_S^*)$ for any given energy harvesting duration α

(ii) In the stage II, the global optimal solution $(\alpha^*, \mathbf{v}_P^*, \mathbf{v}_S^*)$ can be found based on one-dimension search over α

In the stage I, the maximization of the primary secrecy rate is equivalent to maximizing the achievable rate of the PR subject to an alternative upper bound on the achievable rate of ME. Thus, for a given $\alpha = \alpha_0$, $R_{\mathrm{ST}}(\alpha_0)$ is the constant value and the problem (P1) can be transformed into the following problem (P2):

$$\begin{aligned}
&\max_{\mathbf{v}_P,\mathbf{v}_S} \quad \frac{(1-\alpha_0)T}{2}\log_2\left(1 + \frac{P_P|h_{\mathrm{PP}}|^2}{\delta_{\mathrm{PR}}} + \frac{\left|\mathbf{h}_{\mathrm{SPR}}^H\mathbf{v}_P\right|^2}{\left|\mathbf{h}_{\mathrm{SPR}}^H\mathbf{v}_S\right|^2 + \delta_{\mathrm{PR}}}\right) \\
&\text{s.t.} \quad \text{C1: } \frac{(1-\alpha_0)T}{2}\log_2\left(1 + \frac{\left|\mathbf{h}_{\mathrm{SS}}^H\mathbf{v}_S\right|^2}{\left|\mathbf{h}_{\mathrm{SS}}^H\mathbf{v}_P\right|^2 + \delta_{\mathrm{SR}}}\right) \geq r_S \\
&\qquad \text{C2: } \left\|\mathbf{v}_P\right\|^2 + \left\|\mathbf{v}_S\right\|^2 \leq \frac{2\left(\alpha_0\eta P_P \left\|\mathbf{h}_{\mathrm{PST}}\right\|^2 + E_{\mathrm{ST0}}\right)}{1-\alpha_0} \\
&\qquad \text{C3: } \frac{(1-\alpha_0)T}{2}\log_2\left(1 + \frac{P_P|h_{\mathrm{PME}}|^2}{\delta_{\mathrm{ME}}} + \frac{\left|\mathbf{h}_{\mathrm{SME}}^H\mathbf{v}_P\right|^2}{\left|\mathbf{h}_{\mathrm{SME}}^H\mathbf{v}_S\right|^2 + \delta_{\mathrm{ME}}}\right) \leq \Gamma,
\end{aligned} \tag{14}$$

where Γ represents an auxiliary optimization variable to bound the achievable rate of the eavesdropper ME, thus the maximum primary secure rate can be obtained by adjusting value of Γ. The optimal value of Γ^* can be founded by one-dimension search since it is a nonnegative value. Note that the optimization problem (P2) is still nonconvex concerning with beamforming vectors $\mathbf{v}_P$ and $\mathbf{v}_S$.

Considering $\log_2(x)$ is monotonically increasing function of x and defining $\mathbf{H}_{\mathrm{SPR}} = h_{\mathrm{SPR}}h_{\mathrm{SPR}}^H$, $\mathbf{H}_{\mathrm{SME}} = h_{\mathrm{SME}}h_{\mathrm{SME}}^H$, $\mathbf{H}_{\mathrm{SS}} = h_{\mathrm{SS}}h_{\mathrm{SS}}^H$, $\mathbf{V}_P = \mathbf{v}_P\mathbf{v}_P^H$, and $\mathbf{V}_S = \mathbf{v}_S\mathbf{v}_S^H$, the problem (P2) can be denoted as a fractional programming problem, but the objective function is still nonconvex since two optimization variables $\mathbf{V}_P$ and $\mathbf{V}_S$ exist in the numerator and denominator of objective function, respectively. To solve the problem (P2) more effectively, the fractional programming problem can be equivalently reformulated to a convex SDR problem by utilizing Charnes–Cooper transformation [34]. Thus, we let

$$\lambda = \frac{1}{\mathrm{tr}\left(\mathbf{H}_{\mathrm{SPR}}\mathbf{V}_S\right) + \delta_{\mathrm{SR}}}, \tag{15}$$

while defining $\widetilde{\mathbf{V}}_P = \lambda\mathbf{V}_P$ and $\widetilde{\mathbf{V}}_S = \lambda\mathbf{V}_S$, the corresponding SDR of problem (P2) can be rewritten as (P3):

$$\begin{aligned}
&\max_{\mathbf{v}_P,\mathbf{v}_S,\lambda} \quad \mathrm{tr}\left(\mathbf{H}_{\mathrm{SPR}}\widetilde{\mathbf{V}}_P\right) \\
&\text{s.t.} \quad \text{C1: } \mathrm{tr}\left(\mathbf{H}_{\mathrm{SPR}}\widetilde{\mathbf{V}}_S\right) + \lambda\delta_{\mathrm{SR}} = 1, \\
&\qquad \text{C2: } \mathrm{tr}\left(\mathbf{H}_{\mathrm{SS}}\widetilde{\mathbf{V}}_S\right) - \Gamma_S\mathrm{tr}\left(\mathbf{H}_{\mathrm{SS}}\widetilde{\mathbf{V}}_P\right) \geq \lambda\Gamma_S\delta_{\mathrm{SR}}, \\
&\qquad \text{C3: } \mathrm{tr}\left(\widetilde{\mathbf{V}}_P\right) + \mathrm{tr}\left(\widetilde{\mathbf{V}}_S\right) \leq \frac{2\lambda\left(\alpha_0\eta P_P \left\|\mathbf{h}_{\mathrm{PST}}\right\|^2 + E_{\mathrm{ST0}}\right)}{1-\alpha_0} \\
&\qquad \text{C4: } \mathrm{tr}\left(\mathbf{H}_{\mathrm{SME}}\widetilde{\mathbf{V}}_P\right) - \Gamma_e\mathrm{tr}\left(\mathbf{H}_{\mathrm{SME}}\widetilde{\mathbf{V}}_S\right) \leq \lambda\Gamma_e\delta_{\mathrm{ME}} \\
&\qquad \text{C5: } \widetilde{\mathbf{V}}_P \succcurlyeq 0, \widetilde{\mathbf{V}}_S \succcurlyeq 0, \lambda > 0,
\end{aligned} \tag{16}$$

where $\Gamma_S = 2^{2r_S/1-\alpha_0} - 1$ and $\Gamma_e = 2^{2\Gamma/1-\alpha_0} - (P_P|h_{\mathrm{PME}}|^2/\delta_{\mathrm{ME}}) - 1$.

It must be noted that SDR cannot guarantee to derive the optimal solution $(\mathbf{v}_P^*, \mathbf{v}_S^*)$ with rank-one. In the following, the first step is to prove that the rank of optimal $\widetilde{\mathbf{V}}_P^*$ equals to one, and then we propose a method to structure the optimal $\widetilde{\mathbf{V}}_S^*$ with rank-one when the rank of $\widetilde{\mathbf{V}}_S$ is greater than one.

Let θ_1, θ_2, θ_3, and θ_4 represent the Lagrange multipliers, i.e., dual variables, related to constraints C1 to C4 in equation (16), respectively. Thus, the corresponding Lagrange function of problem (P3) can be expressed as

$$\mathscr{L}\left(\widetilde{\mathbf{V}}_P, \widetilde{\mathbf{V}}_S, \theta_1, \theta_2, \theta_3, \theta_4\right) = \mathrm{tr}\left(\xi\widetilde{\mathbf{V}}_P\right) + \mathrm{tr}\left(\psi\widetilde{\mathbf{V}}_P\right) + \rho, \tag{17}$$

where

$$\begin{aligned}
\xi &= \mathbf{H}_{\mathrm{SPR}} - \theta_2\Gamma_S\mathrm{H}_{\mathrm{SS}} - \theta_3\mathbf{I} - \theta_4\mathbf{H}_{\mathrm{SME}}, \\
\psi &= -\theta_1\mathbf{H}_{\mathrm{SPR}} + \theta_2\mathrm{H}_{\mathrm{SS}} - \theta_3\mathbf{I} + \theta_4\Gamma_e\mathbf{H}_{\mathrm{SME}},
\end{aligned} \tag{18}$$

and ρ denotes the residual information that is not related to the proof. According to the definition of Karush–Kuhn–Tucker conditions and Lagrange function of problem (P3), we have

$$\begin{aligned}
\xi^*\widetilde{\mathbf{V}}_P^* &= 0, \\
\psi^*\widetilde{\mathbf{V}}_S^* &= 0.
\end{aligned} \tag{19}$$

Assuming the harvested energy and initial energy are all used for secure beamforming transmission in the third phase, the power constraint C3 in equation (16) is activated with equality, thus the dual variable $\theta_3^* > 0$. Since the transmission channel vectors $\mathbf{H}_{\mathrm{SS}} \succcurlyeq 0$ and $\mathbf{H}_{\mathrm{SME}} \succcurlyeq 0$, we can derive that $\mathrm{rank}(-\theta_2^* \Gamma_S \mathbf{H}_{\mathrm{SS}} - \theta_3^* \mathbf{I} - \theta_4^* \mathbf{H}_{\mathrm{SME}}) = N$. Furthermore, since $\mathrm{rank}(\mathbf{H}_{\mathrm{SPR}}) \le 1$, it follows that $\mathrm{rank}(\xi^*) \ge N - 1$. Based on equation (19), we thus obtain $\mathrm{rank}(\widetilde{\mathbf{V}}_P^*) = 1$.

Define $\kappa^* = -\theta_1^* \mathbf{H}_{\mathrm{SPR}} - \theta_2^* \mathbf{H}_{\mathrm{SS}} - \theta_3^* \mathbf{I} + \theta_4^* \mathbf{H}_{\mathrm{SME}}$, thus we have

$$\psi^* = \kappa^* + 2\theta_2^* \mathbf{H}_{\mathrm{SS}}. \tag{20}$$

Since $\mathbf{H}_{\mathrm{SPR}} \succcurlyeq 0$, $\mathbf{H}_{\mathrm{SS}} \succcurlyeq 0$, and $\mathbf{H}_{\mathrm{SME}} \succcurlyeq 0$, we can obtain that $\mathrm{rank}(-\theta_1^* \mathbf{H}_{\mathrm{SPR}} - \theta_2^* \mathbf{H}_{\mathrm{SS}} - \theta_3^* \mathbf{I}) = N$. Moreover, since $\mathrm{rank}(\mathbf{H}_{\mathrm{SME}}) \le 1$, $\mathrm{rank}(\kappa^*) \ge N - 1$ can be derived:

(i) If $\mathrm{rank}(\kappa^*) = N$, we can obtain $\mathrm{rank}(\psi^*) = N - 1$, thus it follows from equation (19) that $\mathrm{rank}(\widetilde{\mathbf{V}}_S^*) = 1$ and $\widetilde{\mathbf{V}}_S^*$ is equal to aww^H, where $w \in \mathbb{C}^{N \times 1}$ denotes the spanning null space of ψ^* and $a > 0$. Thus, the corresponding optimal value of (P3) is $(\widetilde{\mathbf{V}}_P^*/\lambda^*, \widetilde{\mathbf{V}}_S^*/\lambda^*)$;

(ii) If $\mathrm{rank}(\kappa^*) = N - 1$, we can observe that $\mathrm{rank}(\widetilde{\mathbf{V}}_S^*) > 1$ and thus it requires constructing a new solution with rank-one. First, we obtain the orthonormal basis $u \in \mathbb{C}^{N \times 1}$ of the null base of κ^*, which is defined as $\kappa^* u = 0$ and $\mathrm{rank}(u) = 1$. Then, based on the expression of κ^*, we can further derive that $\mathbf{H}_{\mathrm{SS}} u = 0$. Thus, the optimal solution of $\widetilde{\mathbf{V}}_S^*$ is given by

$$\widetilde{\mathbf{V}}_S^* = buu^H + aww^H, \tag{21}$$

where $b \ge 0$, $\|w\| = 1$, and $w^H u = 0$. Finally, the optimal result of $\widetilde{\mathbf{V}}_S^*$ with rank-one can be rewritten as $\widehat{\mathbf{V}}_S^* = \widetilde{\mathbf{V}}_S^* - buu^H$. Thus, the reconstructed optimal solution for (P3) is $(\widetilde{\mathbf{V}}_P^*/\lambda^*, \widehat{\mathbf{V}}_S^*/\lambda^*)$.

For fixed $\alpha = \alpha_0$, the optimal solutions $(\Gamma^*, \widetilde{\mathbf{V}}_P^*, \widetilde{\mathbf{V}}_S^*)$ can be obtained through one-dimension search Γ based on the following equation:

$$(\Gamma^*, \mathbf{V}_P^*, \mathbf{V}_S^*) = \arg \max_{\alpha = \alpha_0} \text{problem (P3)}, \tag{22}$$

thus, the optimal secure beamforming vectors $(\mathbf{v}_P^*, \mathbf{v}_S^*)$ can be obtained by adopting eigenvalue decomposition (EVD) of $\widetilde{\mathbf{V}}_P^*/\lambda^*$ and $\widetilde{\mathbf{V}}_S^*/\lambda^*$.

In order to obtain the global optimal solution for problem (P1) in the second stage, one-dimension search related to α is then utilized. The optimal solution is chosen from the following equation:

$$(\alpha^*, \Gamma^*, \mathbf{v}_P^*, \mathbf{v}_S^*) = \arg \max_{\alpha \in (0,1)} \text{problem (P1)}. \tag{23}$$

The whole algorithm process can be described in Algorithm 1, which is shown as follows.

3.3. Secure Beamforming Based on Zero-Forcing Rule. This section investigates another secure beamforming solution based on zero-forcing (ZF) rule as a benchmark, in which the primary transmission will not be interfered by other transmissions. Therefore, based on the criterion of ZF rule [35], the beamforming vectors $\mathbf{v}_{S,\mathrm{ZF}}$ and $\mathbf{v}_{P,\mathrm{ZF}}$ for the primary and secondary systems should be in the null space of $\mathbf{h}_{\mathrm{SPR}}^{\perp}$ and $\mathbf{h}_{\mathrm{SS}}^{\perp}$, respectively, i.e., $\mathbf{h}_{\mathrm{SPR}}^H \mathbf{v}_{S,\mathrm{ZF}} = 0$ and $\mathbf{h}_{\mathrm{SS}}^H \mathbf{v}_{P,\mathrm{ZF}} = 0$. Since there exists an eavesdropper in the system to listen the primary's confidential information, the beamforming $\mathbf{v}_{P,\mathrm{ZF}}$ should also be in the null space of $\mathbf{h}_{\mathrm{SME}}^{\perp}$, i.e., $\mathbf{h}_{\mathrm{SME}}^H \mathbf{v}_{P,\mathrm{ZF}} = 0$. In order to be fair in secondary transmission power, we further define $\mathbf{v}_{P,\mathrm{ZF}} = \sqrt{\beta P_{\mathrm{ST}}}\,\widehat{\mathbf{v}}_{P,\mathrm{ZF}}$ and $\mathbf{v}_{S,\mathrm{ZF}} = \sqrt{(1-\beta)P_{\mathrm{ST}}}\,\widehat{\mathbf{v}}_{S,\mathrm{ZF}}$ with $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}} = 1$ and $\widehat{\mathbf{v}}_{S,\mathrm{ZF}}^H \widehat{\mathbf{v}}_{S,\mathrm{ZF}} = 1$, where β represents the power allocation coefficient and $P_{\mathrm{ST}} = 2(\alpha\eta P_P \|\mathbf{h}_{\mathrm{PST}}\|^2 + E_{\mathrm{ST0}})/(1-\alpha)$ denotes the secondary transmission power. Based on equations (13) and (14), the optimization problem based on the ZF rule can be formulated as (P4)

$$\begin{aligned}
\max_{\widehat{\mathbf{v}}_{P,\mathrm{ZF}}, \widehat{\mathbf{v}}_{S,\mathrm{ZF}}} \quad & \frac{(1-\alpha)T}{2} \log_2\left(1 + \frac{P_P |h_{\mathrm{PP}}|^2 + \beta P_{\mathrm{ST}} |\mathbf{h}_{\mathrm{SPR}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}}|^2}{\delta_{\mathrm{PR}}}\right) \\
\text{s.t.} \quad & \mathrm{C1}: \frac{(1-\alpha)T}{2} \log_2\left(1 + \frac{(1-\beta)P_{\mathrm{ST}} |\mathbf{h}_{\mathrm{SS}}^H \widehat{\mathbf{v}}_{S,\mathrm{ZF}}|^2}{\delta_{\mathrm{SR}}}\right) \ge r_S \\
& \mathrm{C2}: \frac{(1-\alpha)T}{2} \log_2\left(1 + \frac{P_P |h_{\mathrm{PME}}|^2}{\delta_{\mathrm{ME}}}\right) \le \Gamma \\
& \mathrm{C3}: \mathbf{h}_{\mathrm{SPR}}^H \widehat{\mathbf{v}}_{S,\mathrm{ZF}} = 0, \mathbf{h}_{\mathrm{SS}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}} = 0, \mathbf{h}_{\mathrm{SME}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}} = 0 \\
& \mathrm{C4}: 0 < \alpha < 1.
\end{aligned} \tag{24}$$

Based on the objective function of the optimization problem (P4), we can observe that the optimal $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}$ should maximize the primary transmission rate under the constraint C3. Thus, the optimal $\mathbf{v}_{P,\mathrm{ZF}}$ can be obtained by utilizing the following optimization problem:

$$\begin{aligned}
\max_{\mathbf{v}_{P,\mathrm{ZF}}} \quad & |\mathbf{h}_{\mathrm{SPR}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}}|^2 \\
\text{s.t.} \quad & \mathbf{h}_{\mathrm{SS}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}} = 0, \mathbf{h}_{\mathrm{SME}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}} = 0.
\end{aligned} \tag{25}$$

Since both the constraint functions in equation (25) include $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}$, we thus can define a new matrix $\mathbf{H}_S = [\mathbf{h}_{\mathrm{SS}}^H; \mathbf{h}_{\mathrm{SME}}^H]$ and the constraint function can be rewritten as $\mathbf{H}_S \widehat{\mathbf{v}}_{P,\mathrm{ZF}} = 0$. To satisfy the new constraint, $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}$ can be obtained by solving the orthogonal value of $\mathbf{H}_S$, which means that $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}$ should be the null space of $\mathbf{H}_S$. To obtain the maximization of $|\mathbf{h}_{\mathrm{SPR}}^H \widehat{\mathbf{v}}_{P,\mathrm{ZF}}|^2$, the optimal $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}^*$ should be chosen the one which is in the direction of the orthogonal projection of $\mathbf{h}_{\mathrm{SPR}}^H$ on to the subspace $\mathbf{H}_S^{\perp}$, where the optimal $\widehat{\mathbf{v}}_{P,\mathrm{ZF}}^*$ is given by

$$\widehat{\mathbf{v}}_{P,\mathrm{ZF}}^* = \frac{\left(\mathbf{I} - \left(\mathbf{H}_S \mathbf{H}_S^H / \|\mathbf{H}_S\|^2\right)\right)\mathbf{h}_{\mathrm{SPR}}}{\left\|\left(\mathbf{I} - \left(\mathbf{H}_S \mathbf{H}_S^H / \|\mathbf{H}_S\|^2\right)\right)\mathbf{h}_{\mathrm{SPR}}\right\|}. \tag{26}$$

Similarly, the optimal $\widehat{\mathbf{v}}_{S,\mathrm{ZF}}^*$ can be derived by analyzing the constraint function $\mathbf{h}_{\mathrm{SPR}}^H \widehat{\mathbf{v}}_{S,\mathrm{ZF}} = 0$ in equation (24), where

Initialize $\alpha = \alpha_0$ and $\Gamma = \Gamma_0$; define $\Gamma_{\max}$ as a large positive real number; $\Delta\alpha$ and $\Delta\tau$ are all small positive real numbers as the iterative steps for one-dimension search
for a given $\alpha = \alpha_0$ **do** S1-S4
S1: given $\Gamma = \Gamma_0$, then solve problem (P3) and derive the optimal solution $(\widetilde{\mathbf{V}}_P^*, \widetilde{\mathbf{V}}_S^*, \lambda^*)$ by utilizing CVX tools
S2: obtain optimal $(\widetilde{\mathbf{V}}_P^*, \widetilde{\mathbf{V}}_S^*)$ through the following procedures
if $\text{rank}(\widetilde{\mathbf{V}}_P^*) = 1$ and $\text{rank}(\widetilde{\mathbf{V}}_S^*) = 1$, **then**
The optimal solution for problem (P3) is $(\widetilde{\mathbf{V}}_P^*/\lambda^*, \widetilde{\mathbf{V}}_S^*/\lambda^*)$
else
Reconstruct an optimal solution $(\widetilde{\mathbf{V}}_P^*/\lambda^*, \widetilde{\mathbf{V}}_S^*/\lambda^*)$ for problem (P3) with $\text{rank}(\widetilde{\mathbf{V}}_P^*) = 1$ and $\text{rank}(\widehat{\mathbf{V}}_S^*) = 1$ based on equation (21)
end if
S3: **let** $\Gamma = \Gamma + \Delta\tau$ when $\Gamma < \Gamma_{\max}$ and then go to S1-S2
S4: **choose** the optimal solution $(\Gamma^*, \mathbf{V}_P^*, \mathbf{V}_S^*)$ from equation (22) and derive optimal secure beamforming vectors $(\mathbf{V}_P^*, \mathbf{V}_S^*)$ by performing EVD
end for
Update $\alpha = \alpha + \Delta\alpha$ and S1-S4
Choose the optimal solution $(\alpha^*, \Gamma^*, \mathbf{v}_P^*, \mathbf{v}_S^*)$ based on equation (23)

Algorithm 1: Optimal secure beamforming design.

$\widehat{\mathbf{v}}_{S,ZF}^*$ should be the null space of $\mathbf{h}_{SPR}^{\perp}$, i.e., $\widehat{\mathbf{v}}_{S,ZF}^*$ belongs to the subspace $\mathbf{h}_{SPR}^{\perp}$. Here, we try to maximize the $|\mathbf{h}_{SS}^H \widehat{\mathbf{v}}_{S,ZF}|^2$ so that more ST's transmission power can be used to transfer primary data to effectively ensure the secure transmission of information in the primary system. Therefore, the optimal $\widehat{\mathbf{v}}_{S,ZF}^*$ can be derived as

$$\widehat{\mathbf{v}}_{S,ZF}^* = \frac{\left(\mathbf{I} - \left(\mathbf{h}_{SPR}\mathbf{h}_{SPR}^H / \|\mathbf{h}_{SPR}\|^2\right)\right)\mathbf{h}_{SS}}{\left\|\left(\mathbf{I} - \left(\mathbf{h}_{SPR}\mathbf{h}_{SPR}^H / \|\mathbf{h}_{SPR}\|^2\right)\right)\mathbf{h}_{SS}\right\|}. \tag{27}$$

According to (24), we can find that the objective function is an increasing function while C1 is a decreasing function with the increase of β and we can obtain the optimal β^* through deriving the upper bound of β. Therefore, the optimal β^* can be expressed as

$$\beta^* = 1 - \delta_{SR}\left(\frac{2^{2r_S/(1-\alpha)T} - 1}{P_{ST}\left|\mathbf{h}_{SS}^H \widehat{\mathbf{v}}_{S,ZF}^*\right|^2}\right). \tag{28}$$

Then, the optimal energy harvesting duration α^* and Γ^* can be derived by adopting one-dimensional search.

4. Simulations and Analyses of Security Transmission Performance

In this section, we will verify the security transmission performance of the primary and transmission efficiencies of the secondary system by comparing the proposed scheme and ZF-based scheme. Unless stated otherwise, we assume that all noise power are normalized to unity, i.e., $\delta_{PR} = \delta_{SR} = \delta_{ME} = 1$. We also consider a scenario where the transmission distance between the PT and PR is 8 m, while the distance between the ST and SR is 3 m. Moreover, the ST is equipped with 4 antennas, and the energy harvesting efficiency is set as $\eta = 0.5$. The transmission channel can be modeled as $h = d^{-\varpi/2}e^{jw}$ with d and $\varpi = 3.5$ denoting the distance and path loss exponent, respectively [36]. The minimum transmission rate of the secondary system and maximal auxiliary optimization variable is set to be $r_S = 0.5$ bit/s/Hz and $\Gamma_{\max} = 1.0$ bit/s/Hz, respectively.

Figure 2 illustrates the secrecy rate of the primary system with respect to the primary transmission power for different initial energies at the ST. In this figure, both the secrecy rates of the primary system with the proposed scheme and ZF scheme are improved with the increase of primary transmission power, respectively. Moreover, the proposed scheme outperforms the ZF scheme in terms of the primary's secrecy rate. With the lower primary transmission power, the superiority of the proposed scheme is obvious and the primary secrecy rates with both schemes are close in high primary transmission power. With the increase of the initial energy at the ST, the secrecy rate gets better as shown in Figure 2 since the more transmission power will be utilized to assist the transmission of the primary signals.

Figure 3 compares the secrecy rates of the primary system with the proposed scheme and ZF scheme against the antenna number at the ST. Obviously, with the increase of the antenna number, the secrecy rates gets better continually since more antennas will result in a higher spatial reuse efficiency. Similarly, the primary secrecy rate is always high for the proposed scheme.

Figure 4 shows the primary secrecy rates with the proposed scheme and ZF scheme against the transmission distance between the PT and ST. From this figure, we can observe that the proposed scheme is superior to the ZF scheme in terms of the primary secrecy rate, regardless the position of the ST. With the increase of the d_{PST}, the primary secrecy rates first become better and then become worse. When the transmission distance d_{PST} is short, the secrecy rates get better with the increase of the d_{PST} because more energy will be harvested for signal transmission and shorter distance for primary signal transferring. However, when the distance d_{PST} is longer, the secrecy rates get worse since the amount of harvested energy will be decreased and more path-loss will result in a negative effect for the ST to process the PT's signal. Furthermore, we can obtain that the optimal positions of the ST are roughly 3m and 4m for the proposed scheme and ZF scheme, respectively.

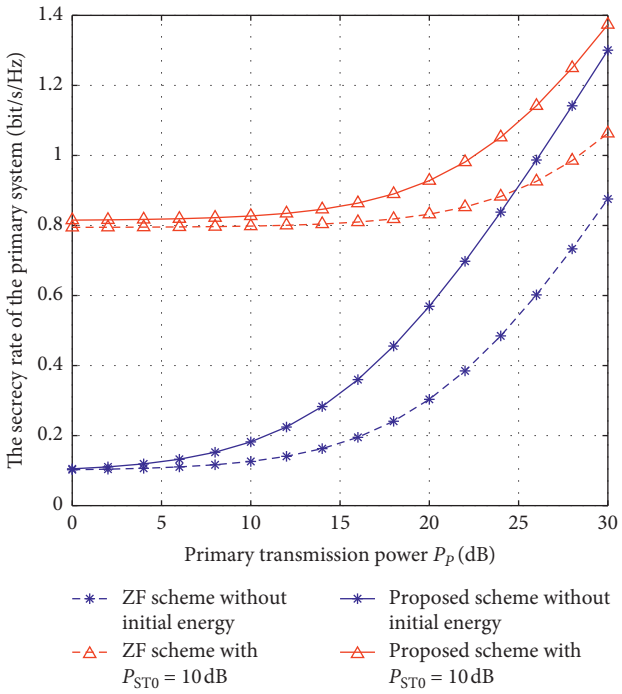


Figure 2: The secrecy rate of the primary system with respect to the primary transmission power P_P for different initial energies at the ST. The antenna number $N = 4$, $d_{PST} = 4$ m, $d_{SPR} = d_{PP}$-d_{PST}, $d_{PME} = d_{PP}$.

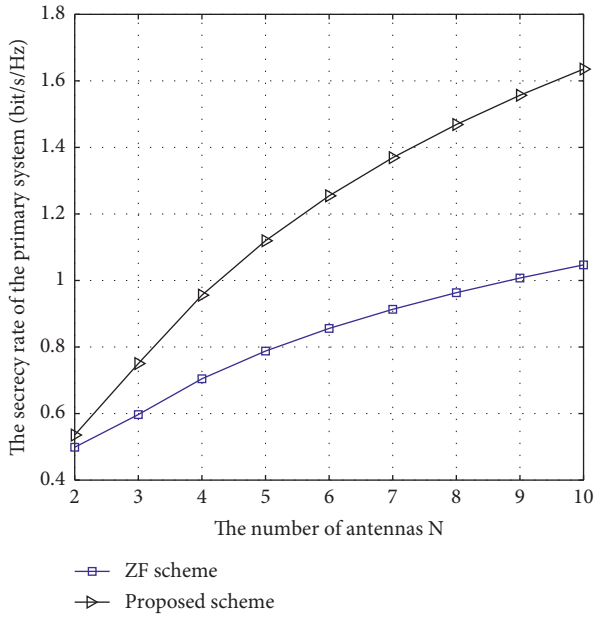


Figure 3: The secrecy rate of the primary system with respect to the number of antenna at the ST. $P_P = 10$ dB, $P_{ST0} = 0$ dB. $d_{PST} = 4$ m, $d_{SS} = 2$ m, $d_{SPR} = d_{PP}$-d_{PST}, $d_{SME} = d_{SPR}$, $d_{PME} = d_{PP}$.

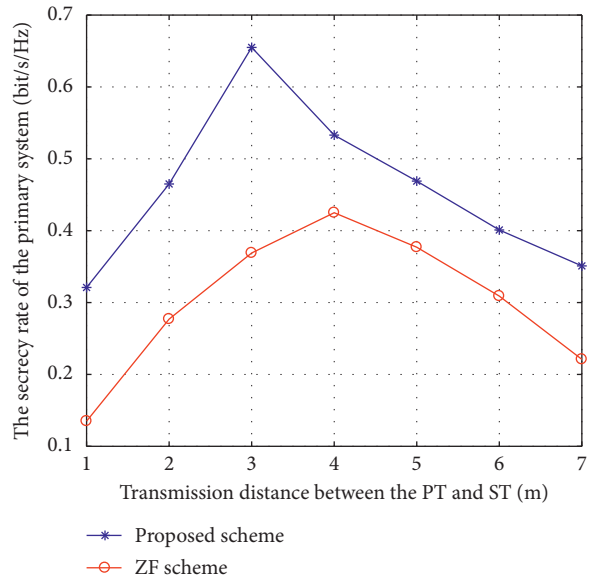


Figure 4: The secrecy rate of the primary system with respect to the distance between the PT and ST. $P_P = 10$ dB, $P_{ST0} = 0$ dB, $d_{SS} = 2$ m, $d_{SPR} = d_{PP}$-d_{PST}, $d_{SME} = d_{SPR}$, $d_{PME} = d_{PP}$. The antenna number $N = 4$.

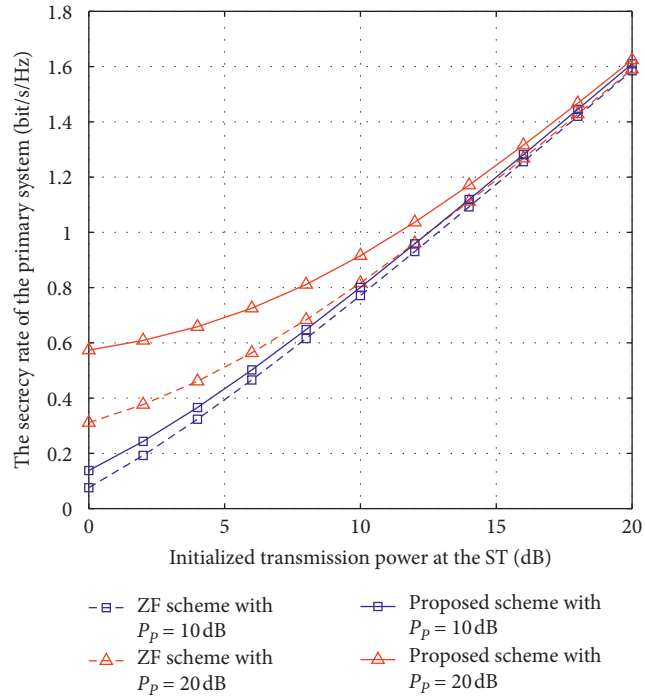


Figure 5: The secrecy rate of the primary system with respect to the initialized transmission power P_{ST0} at the ST for different primary transmission power P_P. $d_{PST} = 4$ m, $d_{SS} = 2$ m, $d_{SPR} = d_{PP}$-d_{PST}, $d_{SME} = d_{SPR}$, $d_{PME} = d_{PP}$. The antenna number $N = 4$.

Figure 5 shows the secrecy rate of the primary system corresponding to the ST's initial energy for different primary transmission power. In this figure, we can observe that the secrecy rates of the primary system with both the schemes are close with the increase of the ST's initial energy, which further illustrates the proposed scheme is superior to the ZF scheme. Specifically, the proposed scheme outperforms the ZF scheme in a lower primary power range. However, in the higher initial primary power range, the gap of the secrecy rates of the primary system between the proposed scheme and the ZF scheme gets small. Therefore, the proposed scheme in this paper is more effective when the initial energy is small.

Figure 6 shows the achievable rate of the secondary system with respect to the primary transmission power.

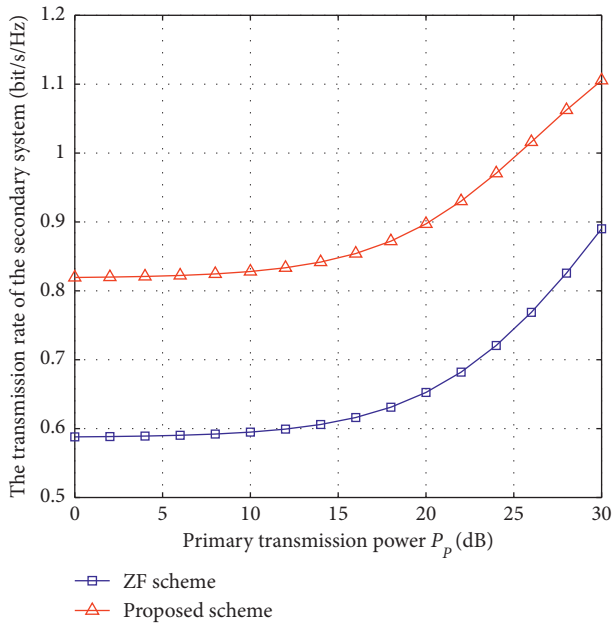


FIGURE 6: The transmission rate of the secondary system with respect to the primary transmission power P_P. $P_{ST0} = 10$ dB. $d_{PST} = 4$ m, $d_{SS} = 2$ m, $d_{SPR} = d_{PP}$-d_{PST}, $d_{SME} = d_{SPR}$, $d_{PME} = d_{PP}$. The antenna number $N = 4$.

From the figure, the throughput of the secondary system with both the scheme is enhanced with the increase of the primary transmission power, which because of more energy will be harvested for the signal transmission. In the meanwhile, the propose scheme outperforms the ZF scheme, which verifies the effectiveness of the proposed scheme.

5. Conclusions

This paper studied the secure transmission problem for the cognitive radio-based IoMT with energy harvesting when the sensitive medical data sent from the PT can be listened by a malicious eavesdropper. For the sake of protecting the security of the sensitive data, we formulate the corresponding optimization problem and propose a novel algorithm for jointly designing the optimal EH duration and secure beamforming vectors to maximizing the primary secrecy transmission rate while ensuring the transmission requirement of the secondary system. In fact, the number of eavesdroppers may usually be more than one, and the proposed scheme still can be utilized to obtain optimized beamforming vectors. The numerical results presents excellent secure transmission performance with the proposed scheme than zero-forcing scheme, which can be implemented into the IoMT devices to effectively protect the security of the sensitive data.

Data Availability

The simulation results based on Matlab 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

The authors thank the Research Foundation of China Postdoctoral Science Foundation under Grant no. 2019M652895, in part by the Research Foundation of Education Department of Hunan Province under Grant no. 18B517, in part by the Teaching Reform Research Project of Hunan University of Science and Engineering under Grant no. XKYJ2018023, and in part by the Construct Program of Applied Characteristic Discipline in Hunan University of Science and Engineering.

References

[1] C. Zhu, V. C. M. Leung, and L. Shu, "Green internet of things for smart world," *IEEE Access*, vol. 3, pp. 2151–2162, 2015.

[2] K. Zhang, J. Ni, K. Yang, J. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[3] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[4] F. Shen, L. Bedogni, and L. Bononi, "A collaborative internet of things architecture for smart cities and environmental monitoring," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 592–605, 2018.

[5] S. Dhingra, R. B. Madda, A. H. Gandomi, M. Patan, and M. Daneshmand, "Internet of things mobile-air pollution monitoring system (IoT-Mobair)," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5577–5584, 2019.

[6] W. Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 579–590, 2019.

[7] W. Tang, J. Ren, K. Deng, and Y. Zhang, "Secure data aggregation of lightweight E-healthcare IoT devices with fair incentives," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8714–8726, 2019.

[8] W. Tang, J. Ren, and K. Zhang, "Efficient and privacy-preserving fog-assisted health data sharing scheme," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, p. 68, 2019.

[9] F. Alsubaei, S. Shiva, and A. Abuhussein, "Security and privacy in the internet of medical things: taxonomy and risk assessment," in *Proceedings of the 42nd IEEE Conference on Local Computer Networks Workshops*, pp. 112–120, Banff, Canada, July 2015.

[10] Federal Communications Comission, *In the Matter of Unlicensed Operation in the TV Broadcast Bands: Second Report and Order and Memorandum Opinion and Order*, FCC, Washington, DC, USA, 2008.

[11] M. Sharma and A. Sahoo, "Stochastic model based opportunistic channel access in dynamic spectrum access networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1625–1639, 2014.

[12] N. Zhang, H. Liang, N. Cheng, Y. Tang, J. W. Mark, and X. S. Shen, "Dynamic spectrum access in multi-channel cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 2053–2064, 2014.

[13] D. Jiang, Y. Wang, C. Yao, and Y. Han, "An effective dynamic spectrum access algorithm for multi-hop cognitive wireless networks," *Computer Networks*, vol. 84, pp. 1–16, 2015.

[14] C. Han, J. Li, Y. Yang, and F. Ye, "Combining solar energy harvesting with wireless charging for hybrid wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 560–576, 2018.

[15] I. Ahmed, M. M. Butt, C. Psomas, and A. Mohamed, I. Krikidis and M. Guizani, Survey on energy harvesting wireless communications: challenges and opportunities for radio resource allocation," *Computer Networks*, vol. 88, pp. 234–248, 2015.

[16] H. Chen, C. Zhai, Y. Li, and B. Vucetic, "Cooperative strategies for wireless-powered communications: an overview," *IEEE Wireless Communications*, vol. 25, no. 4, pp. 112–119, 2018.

[17] K. Tang, R. Shi, and J. Dong, "Throughput analysis of cognitive wireless acoustic sensor networks with energy harvesting," *Future Generation Computer Systems*, vol. 86, pp. 1218–1227, 2018.

[18] Y. Zhang, C. Xu, X. Lin, and S. Shen, "Blockchain-based public integrity verification for cloud storage against procrastinating auditors," *IEEE Transactions on Cloud Computing*, 2019.

[19] Mamta and S. Prakash, "An overview of healthcare perspective based security issues in wireless sensor networks," in *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development*, pp. 870–875, New Delhi, India, 2016.

[20] H. Chen, Y. Li, Y. Jiang, Y. Ma, and B. Ma, "Distributed power splitting for SWIPT in relay interference channels using game theory," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 410–420, 2015.

[21] W. Lu, W. Zhao, S. Hu, B. Liu, B. Li, and Y. Gong, "OFDM based SWIPT for two-way AF relaying network," *IEEE Access*, vol. 6, pp. 73223–73231, 2018.

[22] L. Shi, Y. Ye, R. Q. Hu, and H. Zhang, "Energy efficiency maximization for SWIPT enabled two-way DF relaying," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 755–759, 2019.

[23] Z. Zhang, S. Chen, X. Zhang, and H.-L. Liu, "Outage performance analysis of wireless energy harvesting relay-assisted random underlay cognitive networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2691–2699, 2018.

[24] S. Liu, W. Ejaz, and M. Ibnkahla, "Energy and spectral efficient cognitive radio sensor networks for Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3220–3233, 2018.

[25] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "HealthDep: an efficient and secure deduplication scheme for cloud-assisted eHealth systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4101–4112, 2018.

[26] D. S. Gurjar, H. H. Nguyen, and H. D. Tuan, "Wireless information and power transfer for IoT applications in overlay cognitive radio networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3257–3270, 2019.

[27] H. A. B. Salameh, S. Almajali, M. Ayyash, and H. Elgala, "Spectrum assignment in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1904–1913, 2018.

[28] Y. Huo, M. Xu, X. Fan, and T. Jing, "A novel secure relay selection strategy for energy-harvesting-enabled internet of things," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, pp. 1–18, 2018.

[29] Z. Wang, Z. Chen, B. Xia, J. Luo, and J. Zhou, "Cognitive relaynetworks with energy harvesting and information transfer: design, analysis, and optimization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2562–2576, 2016.

[30] A. Mukherjee, T. Acharya, and M. R. A. Khandaker, "Outage analysis for SWIPT-enabled two-way cognitive cooperative communications," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 9032–9036, 2018.

[31] C. Zhai, J. Liu, and L. Zheng, "Relay-based spectrum sharing with secondary users powered by wireless energy harvesting," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 1875–1887, 2016.

[32] C. Tang, G. Pan, and T. Li, "Secrecy outage analysis of underlay cognitive radio unit over nakagami-," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 609–612, 2014.

[33] X. Chen, J. Chen, and T. Liu, "Secure transmission in wireless powered massive MIMO relaying systems: performance analysis and optimization," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8025–8035, 2016.

[34] W. Wu, B. Wang, Y. Zeng, H. Zhang, Z. Yang, and Z. Deng, "Robust secure beamforming for wireless powered full-duplex systems with self-energy recycling," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10055–10069, 2017.

[35] G. Zhang, I. Krikidis, and B. Ottersten, "Full-duplex cooperative cognitive radio with transmit imperfections," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2498–2511, 2013.

[36] G. Zhang, H. Z. Jorswieck, and B. Ottersten, "Information and energy cooperation in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 9, pp. 2290–2303, 2014.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 5047379, 11 pages
https://doi.org/10.1155/2020/5047379

Research Article

Cryptanalysis and Security Improvement of Two Authentication Schemes for Healthcare Systems Using Wireless Medical Sensor Networks

Jiaqing Mo,[1] **Zhongwang Hu**,[1] **and Yuhua Lin**[2]

[1]*School of Computer Science and Software, Zhaoqing University, Zhaoqing 526061, China*
[2]*Education Technology and Computer Center, Zhaoqing University, Zhaoqing 526061, China*

Correspondence should be addressed to Jiaqing Mo; mojiaqing@126.com

Received 18 October 2019; Accepted 18 January 2020; Published 19 February 2020

Guest Editor: Geethapriya Thamilarasu

Wireless medical sensor networks (WMSNs) play an important role in collecting healthcare data of the remote patient and transmitting them to the medical professional for proper diagnosis via wireless channel. To protect the patient's healthcare data which is private-related and sensitive, some authentication schemes for healthcare systems using WMSN have been proposed to ensure the secure communication between the medical sensors and the medical professional. Since cryptanalyzing the security defects of authenticated protocols is crucial to put forward solutions and propose truly robust protocols, we scrutinize two state-of-the-art authentication protocols using WMSN for healthcare systems. Firstly, we examine Ali et al.'s enhanced three-factor based authentication protocol and show that although it provides a formal proof and a security verification, it still fails to resist offline dictionary guessing attack, desynchronization attack, and privileged insider attack and contains a serious flaw in the password change phase. Secondly, we investigate Shuai et al.'s lightweight and three-factor based authentication protocol and point out that it cannot achieve high security level as they claimed; it is actually subject to offline dictionary guessing attack and privileged insider attack, and it also has a design flaw in the password change phase. In addition, we suggest several countermeasures to thwart these security weaknesses in these two schemes for WMSN and the similar kinds.

1. Introduction

Internet of Things (IoT), which enables a variety of things to connect each other via the Internet or wireless communication, by employing data-collecting devices such as sensors and radio frequency identification (RFID), has a wide range of applications [1, 2]. As an indispensable part of IoT, wireless sensor networks (WSNs) can collect data from specific objects and share them with human beings; thus, WSN is widely applied in many application scenarios, like healthcare service [3, 4], environment monitoring [5], and habitats [6]. Wireless medical sensor network (WMSN) is a popular application of WSN for healthcare systems, in which wearable sensors gather the patient's physiological information such as blood pressure, body temperature, and heart rate and send them to the medical professionals for diagnosis or further treatment [7]. It is obvious that WMSN not only monitors the patient in real-time but also saves his time and money and improves the efficiency of the medical professional. Generally, a typical WMSN mainly includes three entries: a gateway node, sensor nodes, and medical professional. The gateway node (GWN) has powerful computation and ammunition capabilities and plays the role of a communication bridge between the sensors and medical professionals. The sensor nodes, resource-restraint in computation and communication capabilities, are implanted or installed in the patient's body to gather the physiological information and transmit them to medical professionals in the distance with the help of GWN. However, the physiological information of the patient is sensitive, and they are transmitted over insecure wireless channel. If the attacker intercepts and modifies these physiological data, the doctor may make a wrong diagnosis.

Although some measures have been developed to protect the security of WSN at link layer and network layer in IEEE 802.15.4 by IETF [8, 9], it is still necessary to design a robust authentication mechanism in application layer to protect the sensitive sensed data from unauthorized access. That is to say, the identity legitimacy of the medical professional should be verified before accessing the sensor data. In addition, the sensor node to be accessed should be authenticated for the criticality and sensitivity of the sensed data from the patient. Particularly, a session key should be negotiated between the medical professional and the sensor node to secure the real-time access.

Over years, a series of authentication protocols have been proposed for WMSN to protect the transmitted data against unauthorized access from an attacker or a malicious user. We brief the previous schemes related to WMSN. Because of the limitations of wearable sensor's computation and communication capabilities, WMSN authentication schemes concern efficiency and adopt lightweight cryptography operations on the premise of ensuring security. In 2012, Kumar et al. [10] presented an efficient remote user authentication protocol named E-SAP for healthcare applications in WMSN environment and claimed that their scheme is secure against various known attacks. However, He et al. [11] showed that Kumar et al.'s scheme suffers from offline password guessing attack and privileged insider attack, as well as failure to provide user anonymity. In addition, they suggested a robust and efficient anonymous authentication protocol for patient monitoring using WMSN. Unfortunately, both Wu et al. [12] and Li et al. [13] indicated that the protocol in [11] is still vulnerable to some security weaknesses, such as denial of service attack, lack of wrong password detection mechanism, user impersonation attack, senor node capture attack, and offline password guessing attack. As a remedy, they also gave their enhanced protocol, respectively. However, Das et al. [14] observed that Li et al.'s scheme [13] is insecure to withstand sensor node capture attack, privileged insider attack, and lack of user anonymity. Further, they contributed an efficient and secure authentication protocol for WMSN. In the same year, Srinivas et al. [15] described that Wu et al.'s scheme [12] is subject to insider attack, user impersonation attack, and stolen smartcard attack. To thwart these security defects, they devised an efficient authentication scheme using lightweight operations for WMSN. But Wu et al. [16] pointed out that the scheme in [15] is unsuitable for practical deployment owing to security weaknesses like offline password guessing attack, and a lightweight two-factor authentication scheme for healthcare systems using WMSN was introduced to fix these drawbacks.

In 2016, Amin et al. [17] proposed a two-factor anonymous patient monitoring system using hash function in WMSN. The purpose of scheme in [17] is to design a robust and efficient user authentication protocol so as to provide secure data access in WMSN. However, Jiang et al. [18] claimed that scheme in [17] fails to resist stolen mobile device attack and desynchronization attack, as well as suffering from security issue of sensor key exposure. Afterwards, they devised an enhanced protocol. In addition, protocol in [17] was deemed vulnerable to user impersonation attack, offline password guessing attack, known session key temporary information, revelation of secret parameters, and identity guessing attack by Ali et al. [19], and then they proposed an enhanced three-factor authentication protocol to overcome these vulnerabilities. Although Jiang et al. [18] adopted fuzzy verifier technique and asserted that their protocol achieves admirable security properties, we find that their scheme is susceptible to privileged insider attack, denial of service attack, and known session special temporary information attack.

Since elliptical curve cryptography (ECC) can achieve the same symmetric cryptography algorithm (i.e., RSA) security level with faster computation and smaller key size, many authentication protocols have been developed for WMSN on ECC to enhance their security in recent years. In 2016, Hayajneh et al. [20] proposed an authentication protocol for remote patient monitoring with Rabin algorithm and used Tmote sky notes to prove its efficiency. In the same year, Liu and Chung [21] devised a remote user authentication scheme on bilinear pair to facilitate security and privacy protection in wireless healthcare sensor networks and asserted their scheme can resist various known attacks. But, Challa et al. [22] claimed that the protocol in [20] is susceptible to stolen smartcard attack, offline password guessing attack, privileged insider attack, user impersonation attack, and even inappropriate mutual authentication. To improve efficiency and security, they introduced a three-factor authentication protocol using lightweight ECC point multiplications with formal proof. In 2019, to ensure secure communication and privacy-preserving, Xie et al. [23] proposed an efficient and certificateless authentication scheme named CasCP with batch authentication in wireless body area networks. In the same year, Li et al. [2] considered that the protocol in [17] is vulnerable to denial of services (DoS) attack and cannot provide forward secrecy and proposed an ECC-based three-factor authentication protocol using fuzzy commitment and fuzzy verifier techniques to enhance the security of [17].

More recently, Ali et al. [19] analyzed protocol in [17] and showed that their protocol suffers from offline password guessing attack, user impersonation attack, and revelation of secret parameter, and a new three-factor protocol is introduced to resist various attacks. But, in this work, in contrast to their assertions, we examine Ali et al.'s protocol and point out that their scheme is still vulnerable to offline dictionary guessing attack, desynchronization attack, and privileged insider attack and has a flaw in the password change phase. In addition, Shuai et al. [24] in 2019 proposed a lightweight three-factor authentication scheme for patient monitoring using on-body wireless networks and employing one-time hash chain technique and pseudonym identity method to improve its security. The on-body wireless network is actually a WMSN, since the former is like the latter, in which the sensors installed on the patient collect physiological data and transmit them to the doctor or the health professional through GWN for further processing. However, in this paper, we prove that Shuai et al. [24] suffers from three security drawbacks, that is, offline dictionary guessing

attack, privileged insider attack, and flaw in the password change phase.

As two case studies, our analysis shows that a number of WMSN authentication protocols for healthcare systems and the similar kinds are not secure under some provable security models. Furthermore, our cryptanalysis of the two schemes highlights that it is important to pay attention to potential threats when proposing a new authentication protocol.

In brief, our main contributions are summarized as follows.

(1) First, we cryptanalyze Ali et al.'s protocol [19] and reveal that it cannot withstand offline dictionary guessing attack and desynchronization attack and contains a serious flaw in the password change phase.

(2) Second, we cryptanalyze Shuai et al.'s protocol [24] and show that their scheme is vulnerable to offline dictionary guessing attack as [19] and privileged insider attack. In addition, we point out a design flaw in the password change phase of their scheme.

(3) Third, we put forward some effective countermeasures to amend these two schemes and similar authentication protocols with the same defects.

The remainder of this work is organized as follows: In Section 2, we review Ali et al.'s protocol and show their security weaknesses. Shuai et al.'s protocol is reviewed and cryptanalyzed in Section 3. Section 4 puts forward several countermeasures to fix the discovered threats. Finally, conclusion is made in Section 5.

2. Cryptanalysis on Ali et al.'s Protocol

In this section, we briefly review and cryptanalyze Ali et al.'s protocol [19], which is a lightweight three-factor based authentication protocol for healthcare monitoring in WMSN environment. Their scheme consists of five phases: system setup, user registration, login, authentication, and password change. To facilitate description, we list notations in Table 1 and they will be used throughout this work.

Table 1: Notations.

Notation	Description
U_i	i^{th} user
RA	Registration agent
GWN	Gateway node
S_j	j^{th} sensor node
ID_i	The identity of U_i
SID_j	The identity of S_j
PW_i	The password of U_i
BIO_i	U_i's biometrics
$\oplus$	The bitwise XOR operation
$\|\|$	The concatenation operation
$\Longrightarrow$	The secure channel
$\longrightarrow$	The public channel
$h()$	A secure one-way hash function
$H()$	Bio-hash function
$E_k()/D_k()$	The symmetric encryption/decryption function with key k
T_i	The current timestamp, $i = 1, 2, \ldots$
MD	Mobile device of U_i
MID_i	Temporary identity of U_i
ID_g	The identity of GWN
NC_{k0}	Serial number in GWN side
NC_k	Serial number in S_j side

2.1. Review of Ali et al.'s Scheme

2.1.1. System Setup. Firstly, the administrator SA selects identity SID_j for each sensor node and computes $X_{\text{GS}} = h(\text{SID}_j \,||\, X_G)$, $K_j = h(X_{\text{GS}} \,||\, Y_G \,||\, X_G)$, where X_G and Y_G are secret keys of GWN. Afterwards, SA stores $\{X_{\text{GS}}, K_j\}$ in the memory of the sensor node S_j.

2.1.2. User Registration. If the user wants to access the sensor, he must register in the gateway node first.

(1) $U_i \Longrightarrow$ GWN: $\{\text{ID}_i, \text{RPW}_i, F_i\}$

$\text{RPW}_i = h(\text{ID}_i \,||\, \text{PW}_i \,||\, r_i)$, $F_i = H\ (\text{BIO}_i \,||\, r_i)$, where r_i is a random number of U_i.

(2) GWN $\Longrightarrow U_i$: smartcard $= \{A_i, C_i, D_i, \text{DID}_i, H(), h()\}$

$A_i = h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g) \oplus h(\text{RPW}_i \,||\, F_i)$, $C_i = R_g \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $D_i = h(\text{RPW}_i \,||\, R_g \,||\, F_i)$, and $\{\text{DID}_i, C_i\}$ is stored in GWN's database. DID_i is a dynamic identity chosen by GWN and R_g is a random number.

(3) U_i computes $R_n = r_i \oplus h(\text{ID}_i \,||\, \text{PW}_i \,||\, H(\text{BIO}_i))$ and stores it in the smartcard.

2.1.3. Login

(1) U_i inserts his smartcard, inputs ID_i, PW_i and imprints BIO_i, and then the smartcard computes $r_i = R_n \oplus h(\text{ID}_i \,||\, \text{PW}_i \,||\, H(\text{BIO}_i))$, $\text{RPW}_i = h(\text{ID}_i \,||\, \text{PW}_i \,||\, r_i)$, $F_i = H\ (\text{BIO}_i \,||\, r_i)$, $h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g) = A_i \oplus h(\text{RPW}_i \,||\, F_i)$, $R_g = C_i \oplus \text{h}(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, and $D_i' = h(\text{RPW}_i \,||\, R_g \,||\, F_i)$ and checks $D_i' ? = D_i$. If it fails, U_i aborts this login request.

(2) $U_i \longrightarrow$ GWN: $\{\text{DID}_i, M_1, M_2, M_3\}$.

$M_1 = M_i \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $M_2 = E_{h(M_i \| R_g)}(\text{ID}_i \| \text{SID}_j \| T_1 \| A_i)$, $M_3 = h(\text{ID}_i \,||\, \text{SID}_j \,||\, h(\text{RPW}_i \,||\, F_i))$, where M_i is a random nonce and T_1 is the current timestamp.

2.1.4. Authentication

(1) GWN $\longrightarrow S_j$: $\{M_1, M_4, M_5, M_6\}$

GWN computes $M_i = M_1 \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $R_g = C_i \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$ and decrypts M_2 to obtain $(\text{ID}_i \,||\, \text{SID}_j \,||\, T_1 \,||\, A_i)$ using $h(M_i \,||\, R_g)$. If T_1 is not fresh, GWN aborts this session; otherwise, GWN computes $h(\text{RPW}_i \,||\, F_i) = A_i \oplus h(\text{DID}_i \,||\, X_G \,||\, \text{ID}_g)$, $M_3' = h(\text{ID}_i \,||\, \text{SID}_j \,||\, h(\text{RPW}_i \,||\, F_i))$, and checks whether $M_3' ? = M_3$. If it is false, GWN terminates the

session. Otherwise, GWN computes $X_{GS} = h(SID_j \| X_G)$, $K_j = h(X_{GS} \| Y_G \| X_G)$, $M_4 = E_{h(X_{GS}\|K_j)}(ID_g \| ID_i \| M_i \| A_i \| T_3)$, $M_5 = N_i \oplus h(T_3 \| h(RPW_i \| F_i))$, and $M_6 = h(ID_i \| N_i \| T_3 \| ID_g)$.

(2) $S_j \longrightarrow$ GWN: $\{M_7, M_8, T_5\}$

S_j decrypts M_4 to get $(ID_g \| ID_i \| M_i \| A_i \| T_3)$ with key $h(X_{GS} \| K_j)$ and verifies the freshness of T_3. If not, S_j aborts the session. Otherwise, GWN computes $h(RPW_i \| F_i) = M_1 \oplus M_i \oplus A_i$, $N_i = h(T_3 \| h(RPW_i \| F_i)) \oplus M_5$, $M_6' = h(ID_i \| N_i \| T_3 \| ID_g)$ and checks whether $M_6' ?= M_6$. If not, GWN aborts the session. Otherwise, S_j computes $M_7 = V_i \oplus h(M_i \| N_i)$, $SK = h(h(RPW_i \| F_i) \| M_i \| N_i \| V_i)$, and $M_8 = h(SK \| ID_i \| ID_g \| T_5)$.

(3) GWN $\longrightarrow U_i$: $\{M_9, M_{10}\}$

GWN checks the freshness of T_5. GWN aborts the session if T_5 is not fresh. Otherwise, GWN computes $V_i = M_7 \oplus h(M_i \| N_i)$, $SK' = h(h(RPW_i \| F_i) \| M_i \| N_i \| V_i)$, and $M_8' = h(SK' \| ID_i \| ID_g \| T_5)$ and checks whether $M_8' ?= M_8$. If not, GWN aborts the session. Otherwise, GWN computes $C_i^n = R_g \oplus h(DID_i^n \| X_G \| ID_g)$, $M_9 = E_{h(RPW_i\|F_i)}(C_i^n \| N_i \| V_i \| DID_i^n)$, and $M_{10} = h(SK \| C_i^n \| DID_i^n)$ and updates the database with $\{DID_i^n, C_i^n\}$.

(4) U_i decrypts M_9 with $h(RPW_i \| F_i)$ to obtain $(C_i^n \| N_i \| V_i \| DID_i^n)$, computes $SK' = h(h(RPW_i \| F_i) \| M_i \| N_i \| V_i)$, and checks whether $M_{10} ?= h(SK' \| C_i^n \| DID_i^n)$. If yes, U_i replaces (DID_i, C_i) with (DID_i^n, C_i^n). Otherwise, U_i rejects the session.

2.1.5. Password Change. This phase is performed if U_i wants to change his password.

(1) U_i inserts smartcard and keys ID_i, PW_i, and imprints BIO_i, and then the smartcard computes $r_i = R_n \oplus h(ID_i \| PW_i \| H(BIO_i))$, $RPW_i = h(ID_i \| PW_i \| r_i)$, $F_i = H(BIO_i \| r_i)$, $h(DID_i \| X_G \| ID_g) = A_i \oplus h(RPW_i \| F_i)$, $R_g = C_i \oplus h(DID_i \| X_G \| ID_g)$, and $D_i' = h(RPW_i \| R_g \| F_i)$ and compares $D_i' ?= D_i$. If it fails, smartcard aborts the session. Otherwise, the procedure continues.

(2) U_i inputs his new password PW_i^{new}, and the smartcard computes $RPW_i^{new} = h(ID_i \| PW_i^{new} \| r_i)$, $A_i^{new} = A_i \oplus h(DID_i \| X_G \| ID_g) \oplus h(RPW_i^{new} \| F_i)$, and $D_i^{new} = h(RPW_i^{new} \| C_i \oplus h(DID_i \| X_G \| ID_g) \| F_i)$. Finally, smartcard replaces (A_i, D_i) with (A_i^{new}, D_i^{new}).

2.2. Cryptanalysis of Ali et al.'s Protocol. Although Ali et al.'s protocol [19] is equipped with a formal security proof to show that their scheme can withstand various known attacks, it still suffers from some security defects. In this subsection, we prove that their protocol cannot resist offline dictionary guessing attack, desynchronization attack, and privileged insider attack and has a flaw in the password change phase though they tried to fix the security drawbacks in Amin et al.'s scheme. Since it is crucial to depict the capabilities of the adversary in designing a robust authentication protocol in WSN environment, we summarize the adversary model as follows [19, 25–27].

(1) The attacker can intercept, delete, modify, and insert the messages exchanged between the related communication parties over public channel.

(2) The attacker cannot guess the secret key and random number since they are assumed sufficiently large.

(3) The attacker can offline enumerate the user-memorable identities and low-entropy passwords in polynomial time simultaneously.

(4) As far as privileged insider attack is taken into account, the privileged-insider in GWN being an attacker can learn the submitted information by the user during the registration phase of authentication protocol.

(5) When considering whether some multifactor authentication protocol can provide truly multifactor security (i.e., the n factors protocol is secure, even if n-1 factors are compromised), it is reasonable to suppose that (i) the attacker can somehow obtain the lost/stolen smartcard and retrieve the secret information by using side-channel attack [28, 29]. (ii) The attacker can collect the biometrics of the user through malicious device without awareness of victim.

2.2.1. Offline Dictionary Guessing Attack. It is widely regarded that the password-based authentication schemes are prone to password guessing attack [30–32], including online password guessing attack and offline password guessing attack, since the users tend to choose a password that is easy to remember. The online password guessing can be relatively detected by judging whether the time of logins exceeds the threshold. On the contrary, during this guessing attack, the attacker does not need to communicate with related communication parties, and thus the offline password guessing attack is not easily surmounted.

In Ali et al.'s scheme, they claimed their scheme not only can withstand password guessing attack, but also can withstand identity guessing attack. Unfortunately, we prove that their claim is not convincing as they claimed. According to the aforementioned adversary model, we assume that the user's lost/stolen smartcard is obtained by the attacker, and the user's biometrics is also collected by the attacker without awareness of owner, and the attackers can launch offline password guessing attack and offline identity guessing attack simultaneously in terms of item 3 in the adversary model, which we call offline dictionary guessing attack. The offline dictionary guessing attack is conducted to get the user's identity and password by the attacker with the following procedure.

Step 1: the attacker extracts the secret data $\{A_i, D_i, DID_i, H(), h(), R_n\}$ from the smartcard by using methods reported in [28].

Step 2: the attacker selects a candidate pair $(\mathrm{ID}_i^*, \mathrm{PW}_i^*)$ from D_{ID} and D_{PW}, where D_{ID} denotes the identity space and D_{PW} denotes the password space.

Step 3: the attacker computes $r_i^* = R_n \oplus h(\mathrm{ID}_i^* \,||\, \mathrm{PW}_i^* \,||\, H(\mathrm{BIO}_i))$, $\mathrm{RPW}_i = h(\mathrm{ID}_i^* \,||\, \mathrm{PW}_i^* \,||\, r_i^*)$, $F_i^* = H(\mathrm{BIO}_i \,||\, r_i^*)$, $h(\mathrm{DID}_i \,||\, X_G \,||\, \mathrm{ID}_g) = A_i \oplus h(\mathrm{RPW}_i^* || F_i^*)$, $R_g^* = C_i \oplus h(\mathrm{DID}_i \,||\, X_G \,||\, \mathrm{ID}_g)^*$.

Step 4: the attacker checks whether the extracted D_i equals the computed $h(\mathrm{RPW}_i^* \,||\, R_g^* \,||\, F_i^*)$.

Step 5: if it holds, the attacker has found a right pair $(\mathrm{ID}_i, \mathrm{PW}_i)$. Otherwise, the attacker repeats steps 2–4 until the right pair $(\mathrm{ID}_i, \mathrm{PW}_i)$ is found.

For ease of achieving user friendliness, Ali et al.'s scheme [19], like previous schemes [12, 17, 18], provides the password update phase, allowing the users to select their own ID and password and make changes. Generally, the user likes to choose an easy-to-remember identity and password, which are often low-entropy. Thus, this makes sense for the attacker to perform offline dictionary guessing attack by enumerating pairs $(\mathrm{ID}_i, \mathrm{PW}_i)$ in polynomial time. Let $|D_{\mathrm{ID}}|$ and $|D_{\mathrm{PW}}|$ represent the size of D_{ID} and D_{PW}, respectively. In addition, we set T_h and T_H as the execution time of hash function $h()$ and bio-hash function $H()$, respectively. The time complexity of the above attack procedure is $O(|D_{\mathrm{ID}}| * |D_{\mathrm{PW}}| * 4T_h * 2T_H)$. Since T_h and T_H are limited, it is clear that the time required by the attacker to carry out the above attack procedure is linear to $|D_{\mathrm{ID}}| * |D_{\mathrm{PW}}|$. As reported in [33, 34], both the identity space D_{ID} and the password space D_{PW} are rather limited in practice (e.g., $|D_{\mathrm{ID}}| \leq D_{\mathrm{PW}}| \leq 10^6$ [33, 34]), and thus, it is possible for the attacker to guess $(\mathrm{ID}_i, \mathrm{PW}_i)$ within polynomial time. Wang and Wang [35] even pointed out that the time spent on the above guessing attack can be greatly reduced to the level of seconds on an ordinary computer. Therefore, Ali et al.'s protocol [19] is vulnerable to offline dictionary guessing attack.

Based on the aforementioned attack, after the attacker has obtained the user's identity and password, he can impersonate the user to log onto GWN with the smartcard and the collected biometrics. In this regard, Ali et al.'s protocol suffers from user impersonation attack.

2.2.2. Desynchronization Attack. To achieve security features of user anonymity and user untraceability, Ali et al.'s protocol [19] makes use of synchronous update mechanism; that is, GWN updates the dynamic identity DID_i and C_i synchronously with U_i via message $\{M_9, M_{10}\}$. In this way, the attacker cannot trace a particular user by eavesdropping messages over the public channel. However, we point out that the attacker can breach this synchronous mechanism by blocking the last message $\{M_9, M_{10}\}$, leading to failure when the user logs onto GWN the next time. Such attack is illustrated as follows.

In Step 7 of the authentication phase after updating $\{\mathrm{DID}_i^n, C_i^n\}$ in the database, GWN sends message $\{M_9, M_{10}\}$ to U_i, where $M_9 = E_{h(\mathrm{RPW}_i \| F_i)}(C_i^n \| N_i \| V_i \| \mathrm{DID}_i^n)$, $M_{10} = h(\mathrm{SK} \,||\, C_i^n \,||\, \mathrm{DID}_i^n)$ and DID_i^n is a new dynamic identity. Upon receiving the message, U_i will generate a session key and replace $\{\mathrm{DID}_i, C_i\}$ with $\{\mathrm{DID}_i^n, C_i^n\}$. If the malicious attacker blocks this message at the end of authentication process, and the parameters $\{\mathrm{DID}_i, C_i\}$ in the user's smartcard remain unchanged while $\{\mathrm{DID}_i, C_i\}$ on the GWN side have been updated, it means the attacker has broken the dynamic identity synchronization mechanism between GWN and the user by means of blocking messages. As a result, the medical professional can no longer log onto GWN to access data from the sensor on the patient.

2.2.3. Privileged Insider Attack. According to item 4 of the adversary model, a privileged insider of GWN obtains the user's registration request information $\{\mathrm{ID}_i, \mathrm{RPW}_i, F_i\}$, as well as the secret data $\{A_i, C_i, D_i, \mathrm{DID}_i, H(), h()\}$ on the smartcard before GWN sent the smartcard to the user. With this information, he launches a privileged insider attack as follows.

Step 1: he eavesdrops the messages $\{M_1, M_4, M_5, M_6\}$ and $\{M_9, M_{10}\}$ from the public channel.

Step 2: then, he decrypts M_9 using decryption key $h(\mathrm{RPW}_i \,||\, F_i)$ to obtain N_i and V_i.

Step 3: further, he acquires M_i by computing $M_i = h(\mathrm{RPW}_i \,||\, F_i) \oplus M_1 \oplus A_i$.

Step 4: finally, with the known parameters $h(\mathrm{RPW}_i \,||\, F_i)$, M_i, N_i, V_i, the attacker can compute the session key $\mathrm{SK} = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, M_i \,||\, N_i \,||\, V_i)$.

Therefore, Ali et al.'s scheme suffers from privileged insider attack.

2.2.4. Flaw in Password Change Phase. In Ali et al.'s protocol, they provide a password change phase to allow users to freely change the password locally. However, our scrutiny reveals that their password change phase has a fatal flaw which will prevent the user from logging onto GWN. In their scheme, before changing the password, the user is asked to input his identity and old password and imprint his biometrics. If the identity legitimacy of the user is verified by the smartcard, the user is allowed to enter a new password to update the old one. Then, the smartcard computes $\mathrm{RPW}_i^{\mathrm{new}} = h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, r_i)$, $A_i^{\mathrm{new}} = A_i \oplus h(\mathrm{DID}_i \,||\, X_q \,||\, \mathrm{ID}_g) \oplus h(\mathrm{RPW}_i^{\mathrm{new}} \,||\, F_i)$, $D_i^{\mathrm{new}} = h(\mathrm{RPW}_i^{\mathrm{new}} \,||\, C_i \oplus h(\mathrm{DID}_i \,||\, X_G \,||\, \mathrm{ID}_g) \,||\, F_i))$. At last, the smartcard replaces $\{A_i, D_i\}$ with $\{A_i^{\mathrm{new}}, D_i^{\mathrm{new}}\}$. Note that R_n has not been updated with the new password $\mathrm{PW}_i^{\mathrm{new}}$. Thereafter, if the user wants to log onto GWN, he enters ID_i, $\mathrm{PW}_i^{\mathrm{new}}$ and imprints BIO_i, and the smartcard computes $r_i' = R_n \oplus h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, H(\mathrm{BIO}_i))$. It is evident that $r_i' \neq r_i$, since $\mathrm{PW}_i^{\mathrm{new}} \neq \mathrm{PW}_i$. Accordingly, because the calculation of D_i is related to r_i, the computed D_i is not equal to the stored D_i in the smartcard. For this reason, the legal user is always rejected from logging onto GWN once he changed his password. Thus, Ali et al.'s protocol suffers with a serious flaw in the password change phase.

3. Cryptanalysis on Shuai et al.'s Protocol

In this section, we review and cryptanalyze Shuai et al.'s protocol [24] proposed in 2019, which is an anonymous authentication scheme for remote patient monitoring. To achieve some desirable security attributes, their scheme employs pseudonym identity method to preserve user anonymity and adopts one-time hash chain technique to achieve forward secrecy. The serial number technique is also used to resist desynchronization attack. Furthermore, they conduct an informal security analysis to show that their scheme is secure against various attacks. However, in the following section, we find that their scheme is susceptible to offline dictionary guessing attack; that is, their protocol fails to provide truly a three-factor security. On the other hand, we show that their protocol is suspected to privileged insider attack.

3.1. Review of Shuai et al.'s Scheme. We will concisely review Shuai et al.'s scheme. Their protocol involves initialization phase, registration phase, login phase, authentication and key agreement phase, and password change phase.

3.1.1. Initialization Phase. The RA performs this phase offline. RA chooses two random numbers ID_g and K as the identity and master secret key to GWN, respectively. Next, RA chooses a collision-resistant cryptographic hash function $h()$ for all communication participants. Finally, RA chooses a unique identity SID_j for each wearable sensor node S_j and stores SID_j into S_j's memory.

3.1.2. Registration Phase. This phase consists of two points, that is, user registration phase and wearable sensor node registration phase.

(1) User registration

(*i*) U_i => RA: $\{ID_i, A_i\}$
The user U_i inputs his ID_i, PW_i, and imprints biometrics BIO_i to mobile device MD. Thereafter, MD computes $\text{Gen}(BIO_i) = (R_i, P_i)$, $A_i = h(PW_i \| R_i \| a_i)$, where Gen is a probabilistic generation procedure, R_i is a secret random key, P_i is an auxiliary string, and a_i is a random secret value generated by U_i.

(ii) RA => U_i: $\{MID_i, B_i, C_i, K_{GU}\}$
RA chooses three nonces b_i, r_1, r_2 and sets $K_{GU} = r_1$, $MID_i = MID_{i0} = r_2$, $MID_{i1} = \text{null}$. Afterwards, RA calculates $B_i = A_i \oplus h(ID_i \| K \| b_i)$, $C_i = h(h(ID_i \| K \| b_i) \| A_i)$, stores $\{ID_i, MID_{i0}, MID_{i1}, b_i, K_{GU}\}$ into the user information table, and copies this table to GWN.

(iii) U_i calculates $D_i = h(ID_i \| PW_i \| R_i) \oplus a_i$ and then stores $\{D_i, P_i\}$ into the MD's memory. Finally, MD contains secret data $\{MID_i, B_i, C_i, D_i, K_{GU}, P_i\}$.

(2) Wearable sensor node registration phase

(*i*) S_j => RA: $\{SID_j\}$.

(ii) RA => S_j: $\{K_{GS}, NC_K\}$.
RA chooses a random nonce K_{GS} and set $NC_K = NC_{K0} = 0$, and then RA stores $\{SID_j, K_{GS}, NC_{K0}\}$ into the sensor node information table and copies it to GWN.

(iii) On receipt of the message, S_j stores $\{K_{GS}, N_{CK}\}$ into its memory.

3.1.3. Login Phase. U_i keys his ID_i, PW_i, and imprints his biometrics BIO_i^* to MD, and MD computes $R_i^* = \text{Rep}(BIO_i^*, P_i)$, $a_i^* = D_i \oplus h(ID_j \| PW_i \| R_i^*)$, $A_i^* = h(PW_i \| R_i \| a_i^*)$, $h(ID_i \| K \| b_i) = B_i \oplus A_i^*$, $C_i^* = h(h(ID_i \| K \| b_i) \| A_i^*)$ and checks whether C_i^* equals the stored C_i. If it is false, MD aborts the session. Otherwise, MD chooses a random nonce R_1 and the current timestamp T_1, computes $MS_1 = (R_1 \| SID_j) \oplus h(MID_i \| h(ID_i \| K \| b_i) \| K_{GU})$, $V_1 = h(ID_i \| R_1 \| h(ID_i \| K \| b_i) \| MID_i \| K_{GU} \| T_1)$. Finally, U_i sends message $\{MID_i, MS_1, V_1, T_1\}$ to GWN.

3.1.4. Authentication and Key Agreement Phase

(1) On receiving the login request, GWN checks the freshness of timestamp T_1. If not, GWN rejects the request. Otherwise, the subsequent operations of GWN are divided into three cases.

Case 1: If $MID_i = MID_{i0}$, GWN extracts $\{ID_i, b_i, K_{GU}, MID_{i1}\}$ from the user information table in light of MID_i and then checks whether the one-time hash chain K_{GU} is updated.

(*i*) If $MID_{i1} = \text{NULL}$, it means that K_{GU} has been updated. GWN computes $(R_1^* \| SID_j) = MS_1 \oplus h(MID_{i0} \| h(ID_i \| K \| b_i) \| K_{GU})$, $V_i^* = h(ID_i \| R_1^* \| h(ID_i \| K \| b_i) \| MID_{i0} \| K_{GU} \| T_1)$, and checks whether $V_i^* = V_i$ holds. If not, GWN aborts the session. Otherwise, GWN chooses a new pseudonym identity MID_{i0}^* and sets $MID_{i1} = MID_{i0}$, $MID_{i0} = MID_{i0}^*$.

(ii) If $MID_{i1} \neq \text{NULL}$, it indicates K_{GU} is not updated in the last session. GWN computes $K_{GU}^* = h(K_{GU})$, $(R_1^* \| SID_j) = MS_i \oplus h(MID_{i0} \| h(ID_i \| K \| b_i) \| K_{GU}^*)$ and $V_i^* = h(ID_i \| R_1^* \| h(ID_i \| K \| b_i) \| MID_{i0} \| K_{GU} \| T_1)$ and checks whether V_i^* equals V_i. If not, GWN aborts this session. Otherwise, GWN generates a new random pseudonym identity MID_{i0}^* and sets $MID_{i1} = MID_{i0}$, $MID_{i0} = MID_{i0}^*$, $K_{GU} = K_{GU}^*$.
Case 2: If $MID_i = MID_{i1}$, GWN extracts $\{ID_i, b_i, K_{GU}\}$, computes $(R_1^* \| SID_j) = MS_1 \oplus h(MID_{i1} \| (h(ID_i \| K \| b_i) \| K_{GU})$, $V_i^* = h(ID_i \| R_1^* \| h(ID_i \| K \| b_i) \| MID_{i1} \| K_{GU} \| T_1)$, and verifies $V_i^* \stackrel{?}{=} V_i$. If not, GWN aborts this session. Otherwise, GWN selects a new random pseudonym identity MID_{i0}^* and sets $MID_{i0} = MID_{i0}^*$.
Case 3: If $MID_{i1} \neq MID_{i0}$ and $MID_i \neq MID_{i1}$, GWN aborts the session.

(2) GWN $\longrightarrow S_j$: $\{MS_2, V_2, NC_{k0}\}$

GWN chooses a random nonce R_2 and computes $MS_2 = (R_1 \| R_2 \| ID_i \| ID_g) \oplus h(K_{GS} \| SID_k \| NC_{k0})$, $V_2 = h(ID_i \| ID_g \| R_1 \| R_2 \| K_{GS} \| NC_{k0})$. Thereafter, GWN updates K_{GS} and NC_{k0} with $K_{GS} = h(K_{GS} \| SID_j)$ and $NC_{k0} = NC_{k0} + 1$, respectively.

(3) $S_j \longrightarrow$ GWN: $\{MS_3, V_3\}$

Upon receiving the message from GWN, S_j checks whether $1 \leq NC_{k0} - NC_k \leq N$ holds, where N is a threshold. If it is false, S_j aborts the session. Otherwise, after setting $K^*_{GS} = K_{GS}$, S_j computes $N-1$ times $K^*_{GS} = h(K^*_{GS} \| SID_j)$. If $N = 1$, S_j will not execute the above hash operation. Then, S_j computes $(R_1 \| R_2 \| ID_i \| ID_g) = MS_2 \oplus h(K^*_{GS} \| SID_j \| (NC_{k0} - 1))$, $V^*_2 = h(ID_i \| ID_g \| R_1 \| R_2 \| K^*_{GS} \| (NC_{k0} - 1))$, and verifies $V^*_2 ? = V_2$. If it is true, S_j sets $K_{GS} = h(K^*_{GS} \| SID_j)$ and $NC_k = NC_{k0}$. Then, S_j generates a random number R_3 and computes $SK = h(ID_i \| ID_g \| SID_k \| R_1 \| R_2 \| R_3)$, $MS_3 = R_3 \oplus h(K_{GS} \| SID_j \| NC_k)$, $V_3 = h(SID_j \| ID_i \| SK \| R_3 \| NC_k)$, and transmits $\{MS_3, V_3\}$ to GWN.

(4) GWN $\longrightarrow U_i$: $\{MS_4, V_4\}$

Upon receiving the message from S_j, GWN computes $R^*_3 = MS_3 \oplus h(K_{GS} \| SID_j \| NC_{k0})$, $SK = h(ID_i \| ID_g \| SID_j \| R_1 \| R_2 \| R^*_3)$, $V^*_3 = h(SID_j \| ID_i \| SK \| R^*_3 \| NC_{k0})$, and verifies $V^*_3 ? = V_3$. If it is false, GWN aborts the session. Otherwise, GWN computes $MS_4 = (R_2 \| R_3 \| ID_g \| MID_{i0}) \oplus h(R_1 \| h(ID_i \| K \| b_i) \| K_{GU} \| MID_{i1})$, $V_4 = h(ID_i \| SID_j \| SK \| R_2 \| MID_{i0})$, and sends $\{MS_4, V_4\}$ to U_i.

(5) $U_i \longrightarrow$ GWN: $\{V_5\}$

Upon receiving the message, U_i computes $(R_2 \| R_3 \| ID_g \| MID_{i0}) = MS_4 \oplus h(R_1 \| h(ID_i \| K \| b_i) \| K_{GU} \| MID_i)$, $SK = h(ID_i \| ID_g \| SID_j \| R_1 \| R_2 \| R_3)$, $V^*_4 = h(ID_i \| SID_j \| SK \| R_2 \| MID_{i0})$, and verifies $V^*_4 ? = V_4$. If it is false, U_i aborts the session. Otherwise, U_i computes $V_5 = h(ID_i \| ID_g \| SID_j \| MID_{i0} \| SK)$ and sets $K_{GU} = h(K_{GU})$ and $MID_i = MID_{i0}$. After that, U_i sends $\{V_5\}$ to GWN.

(6) GWN

Upon receiving $\{V_5\}$, GWN computes $V^*_5 = h(ID_i \| ID_g \| SID_j \| MID_{i0} \| SK)$ and verifies $V^*_5 ? = V_5$. If it is false, GWN aborts the session. Otherwise, GWN sets $K_{GU} = h(K_{GU})$ and $MID_{i1} = NULL$ and believes that U_i has shared a session key with S_j.

3.1.5. Password Change Phase. U_i inputs ID_i, PW_i and imprints BIO_i to mobile device MD. Then, MD computes $R^*_i = Rep(BIO_i, P_i)$, $a^*_i = D_i \oplus h(ID_i \| PW_i \| R^*_i)$, $A^*_i = h(PW_i \| R_i \| a^*_i)$, $h(ID_i \| K \| b_i) = B_i \oplus A^*_i$, $C^*_i = h(h(ID_i \| K \| b_i) \| A^*_i)$, and compares C^*_i with the stored C_i. If it is true, MD rejects the password change request. Otherwise, MD allows U_i to input a new password PW^{new}_i and computes $A^{new}_i = h(PW^{new}_i \| R_i \| a_i)$, $B^{new}_i = h(ID_i \| K \| b_i) \oplus A^{new}_i = B_i \oplus A_i \oplus A^{new}_i$, and $C^{new}_i = h(h(ID_i \| K \| b_i) \oplus A^{new}_i)$. Finally MD updates $\{B_i, C_i\}$ with $\{B^{new}_i, C^{new}_i\}$.

3.2. Cryptanalysis on Shuai et al.'s Scheme. Despite armed with three factors and formal security proof, Shuai et al.'s protocol [24] suffers from offline dictionary guessing attack and privileged insider attack and contains a serious design flaw in the password change phase.

3.2.1. Offline Dictionary Guessing Attack. Suppose the attacker has obtained the lost/stolen mobile device and extracted the secret data $\{MID_i, B_i, C_i, D_i, K_{GU}, P_i\}$ from it; meanwhile, he has collected biometrics BIO_i of the medical professional via a malicious terminal; the attacker can mount an offline dictionary guessing attack as follows.

Step 1: computes $R^*_i = Rep(BIO_i, P_i)$;

Step 2: chooses a pair (ID^*_i, PW^*_i) from the dictionary space DID and DPW, respectively.

Step 3: computes $a^*_i = D_i \oplus h(ID^*_i \| PW^*_i \| r^*_i)$, $A^*_i = h(PW^*_i \| r^*_i \| A^*_i)$, $h(ID^*_i \| K \| b_i)^* = B_i \oplus A^*_i$, $C^*_i = h(h(ID^*_i \| K \| b_i)^* \| A^*_i)$, where D_i and B_i are from the mobile device.

Step 4: verifies the correctness of (ID^*_i, PW^*_i) pair by checking whether the computed C^*_i equals the stored C_i. If it holds, the attacker has found the correct value of (ID^*_i, PW^*_i). Otherwise, the attacker repeats steps 2–4 until $C^*_i = C_i$.

It is clear that the time complexity of the above attack is $O(|D_{ID}| * |D_{PW}| * 3T_h)$, where T_h is the execution time of hash function. As analyzed in Section 2.2.1, such attack is quite efficient.

3.2.2. Privileged Insider Attack. Assume privileged insider of RA being an attacker, it is easy for him to know the registration information $\{ID_i, A_i\}$ during the user registration phase. Moreover, he also can learn $\{ID_i, MID_{i0}, MID_{i1}, b_i, K_{GU}\}$ from the user information table and the registration reply message $\{MID_{i1}, B_i, C_i, K_{GU}\}$ from the side of RA and mount a privileged insider attack. The similar attacks have been discussed in [14, 36–38]. Using these information, the attacker can reveal the session key with the following procedure.

Step 1: computes $h(ID_i \| K \| b_i)^* = A_i \oplus B_i$.

Step 2: intercepts the user's login request message $\{MID_i, MS_1, V_1, T_1\}$ and GWN's reply message $\{MS_4, V_4\}$ from the public channel.

Step 3: acquires R_1 and SID_j by computing $(R_1 \| SID_j)^* = MS_1 \oplus h(MID_i \| h(ID_i \| K \| b_i)^* \| K_{GU})$.

Step 4: acquires R_2, R_3, ID_g, and MID_{i0}, by computing $(R_2 \| R_3 \| ID_g, \| MID_{i0})^* = MS_4 \oplus h(R_1 \| h(ID_i \| K \| b_i)^* \| K_{GU} \| MID_i)$.

Step 5: computes the session key $SK = h(ID_i \| ID_g \| SID_j \| R_1 \| R_2 \| R_3)$.

With the session key, the attacker can decrypt all the messages between the user and the sensor. In this way, the patient's sensitive physiological information is exposed to the attacker. Therefore, Shuai et al.'s scheme fails to resist privileged insider attack.

3.2.3. Flaw in Password Change Phase. For ease of the password change phase, Shuai et al.'s scheme also provides the password change phase for U_i to change his password locally without contacting the RA. Unfortunately, similar to Ali et al.'s scheme, there is a serious security flaw in their password change phase which prevents the users who change their password from being able to log onto GWN again. Before allowing the user to change the password, the MD verifies his identity legitimacy based on the identity ID_i, password PW_i, and biometrics information BIO_i provided by the user. If the user is legitimate, MD allows U_i to input his new password $\mathrm{PW}_i^{\mathrm{new}}$. However, this password change phase only updates B_i and C_i stored on the mobile device according to the new password and does not update D_i with the new password, which is used to recover the secret random number a_i of U_i during the login phase. The user either writes the secret random number a_i on a paper or bears it in mind or updates D_i with the new password. Thus, if he intends to recover a_i by computing $a_i = D_i \oplus h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, R_i^*)$ when he logs onto GWN, he will fail because the previous PW_i is different from the new $\mathrm{PW}_i^{\mathrm{new}}$, and $h()$ is a collision-resistant function, which causes the computed value of $h(\mathrm{ID}_i \,||\, \mathrm{PW}_i \,||\, R_i^*)$ and $h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, R_i^*)$ not to be equal. As a result, the user who has changed his password will be rejected by MD when he intends to log onto GWN again. What is worse, the user can no longer change the password in the future, because MD also needs to verify the legitimacy of the user by recovering the user's secret random number a_i before changing his password.

4. Countermeasures

In order to address the security weaknesses in Ali et al.'s protocol and Shuai et al.'s protocol, we provide several possible countermeasures in this section.

4.1. Countermeasures to Offline Dictionary Guessing Attack. Our previous analysis shows that neither Ali et al.'s scheme nor Shuai et al.'s Scheme can provide truly three-factor security; that is, the attacker can launch an offline dictionary guessing attack to acquire the user's identity and password if he obtains the user's smartcard (or mobile device) and biometrics somehow. The root cause of this attack described above is that the password verifier $D_i = h(\mathrm{RFW}_i \,||\, R_g \,||\, F_i)$ of Ali et al.'s protocol and $C_i = h(h(\mathrm{ID}_i \,||\, K \,||\, b_i) \,||\, A_i)$ are stored in a smartcard (mobile device). Consequently, if the smartcard is obtained by the attacker, he will try to make a breach in the password verifier for offline dictionary guessing attack.

To thwart this security weakness without radical improvement while keeping usability, a feasible countermeasure is to utilize "fuzzy verifier" technique [25]. In the following, taking Ali et al.'s protocol as a case study to show how to integrate fuzzy verifier, we revise the password verifier D_i as $D_i = h(h(\mathrm{RPW}_i \,||\, R_g \,||\, F_i) \bmod n)$ during the user registration phase, where n represents the space size of $(\mathrm{ID}_i, \mathrm{PW}_i)$ pair. If the attacker has obtained the user's smartcard and biometrics, he picks up a pair $(\mathrm{ID}_i^*, \mathrm{PW}_i^*)$ from D_{ID} and D_{PW} to perform offline dictionary guessing attack as described in Section 2.2.1. However, it is hard for the attacker to find a correct pair $(\mathrm{ID}_i, \mathrm{PW}_i)$ since there are $(|D_{\mathrm{ID}}| * |D_{\mathrm{PW}}|)/n \approx 2^{32}$ candidates of $(\mathrm{ID}_i, \mathrm{PW}_i)$ pair (suppose $n = 2^8$, $|D_{\mathrm{ID}}| = |D_{\mathrm{PW}}| = 2^6$ [25, 33]). Someone may question if the attacker will just pick up an incorrect pair of $(\mathrm{ID}_i, \mathrm{PW}_i)$ but can satisfy $D_i = h(h(\mathrm{RPW}_i \,||\, R_g \,||\, F_i) \bmod n)$. The probability of such an event is $1/2^8$. Moreover, if the user is asked to enter the old/new password twice, and the hash function $h()$ responds as a random oracle, the probability will greatly reduce to $(1/2^8)^2 = 1/2^{16}$ [25, 33, 34]. Therefore, the fuzzy verifier that provides adequate candidate can effectively prevent the attacker from mounting offline dictionary guessing attack successfully. In addition, the effectiveness of fuzzy verifier technique has been discussed and verified in Section V-B of [24], and the interested readers can refer to it for more information.

4.2. Countermeasures to Desynchronization Attack. We have demonstrated that Ali et al.'s protocol is insecure against desynchronization attack in Section 2.2.2. Specifically, to provide user anonymity and untraceability, GWN chooses a new dynamic identity DID_i^n, computes the corresponding C_i^n, and stores $\{\mathrm{DID}_i^n, C_i^n\}$ in its database. Meanwhile, to keep consistency in the next login, the user needs to update $\{\mathrm{DID}_i^n, C_i^n\}$ in the smartcard simultaneously. However, Ali et al.'s protocol only considers the case where all messages in the ideal situation are successfully received by the receiver. If the attacker blocks the message $\{M_9, M_{10}\}$ from the GWN to the user to break the consistency in the authentication process, the authenticated parameters $\{\mathrm{DID}_i^n, C_i^n\}$ are made to be different between GWN and the user U_i, which means U_i could not log onto GWN ever since.

To cope with such an attack, an effective countermeasure is to avoid updating the user dynamic identity DID_i simultaneously on both sides of communication parties. That is, during the authentication phase, GWN chooses a new dynamic identity DID_i^n for U_i, but does not need to save it to the database. After decrypting M_9, U_i conceals DID_i^n with the new random number M_i and other information generated in each login, stores it in the smartcard, and restores DID_i^n on the next login. If message $\{M_9, M_{10}\}$ is blocked, on the one hand, the attacker cannot obtain the new DID_i^n because M_9 is encrypted; on the other hand, U_i does not update $D\mathrm{ID}_i$ in the smartcard since he has not received $\{M_9, M_{10}\}$. When U_i logs onto GWN next time, GWN can still recover M_i with the stored $D\mathrm{ID}_i$ instead of DID_i^n. In this way, although the attacker attempts to break the synchronization, he will not succeed because the dynamic identity information of the user has not been saved in GWN, and GWN will perform the subsequent procedure regardless of whether $\{M_9, M_{10}\}$ is blocked or not. Hence, the desynchronization attack is

thwarted effectively. It is worth noting that we only give the main idea of the measure, not a complete scheme, because the detailed solution requires a long paper. In addition, their user registration phase and the password change also need to be revised correspondingly, and we omitted them due to the space constraints.

4.3. Countermeasures to Privileged Insider Attack. Our aforementioned analysis shows that both of the two schemes suffer from privileged insider attack. The root cause is that to improve the computation efficiency, they use lightweight operations based on hash function and random numbers to generate the session key, which makes the leakage of a small amount of secret data easily lead to the leakage of other secret data. To thwart this attack, the public-key operations such as modular exponentiation or elliptic curve point multiplication should be adopted in their scheme [31]. We take the GWN and sensor side as the server side and keep the user as the client side; according to [31], modular exponentiation operation should be performed at least twice on the server side. Take Ali et al.'s scheme as an example and use elliptic curve point multiplication; without requiring radical improvement, the main idea of overcoming privileged insider attack during the login and authentication phase is sketched as follows.

Step 1: after generating the random nonce M_i in the login phase, U_i computes $W_1 = M_i P$ and sends the message containing W_1 to GWN. P is a generator in elliptic curve group over a finite field.

Step 2: because GWN does not need to participate in negotiating session key, GWN sends the message containing W_1 to S_j after the user's identity legitimacy verification is passed.

Step 3: if the legitimacy authentication of GWN is passed, the sensor S_j selects the random number V_i and calculates $W_2 = V_i P$ and computes the session key $\mathrm{SK} = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, W_1 \,||\, W_2 \,||\, V_i W_1) = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, M_i P \,||\, V_i P \,||\, V_i M_i P)$. Afterwards, S_j sends a message containing W_2 to U_i via GWN.

Step 4: if the legitimacy of GWN and S_j is ensured, U_i computes the session key $\mathrm{SK} = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, W_1 \,||\, W_2 \,||\, M_i W_2) = h(h(\mathrm{RPW}_i \,||\, F_i) \,||\, M_i P \,||\, V_i P \,||\, M_i V_i P)$.

If the attacker eavesdrops W_1 and W_2 from the public channel and intends to find M_i and V_i from W_1 and W_2, respectively, it is infeasible since he has to resolve elliptic curve discrete logarithm problem [2]; and if he intends to compute $M_i V_i P$ from W_1 and W_2, it is also impossible since he faces the hardness of elliptic curve computational Diffie-Hellman problem [2].

4.4. Countermeasures to Flaw in Password Change Phase. As we have analyzed before, both Ali et al.'s scheme and Shuai et al.'s scheme contain serious flaws in their password change phase which renders the user unable to log onto GWN again after changing his password. The reason is that none of their password change phase are designed to recover the secret random number for login. Thus, the countermeasures to fix these design flaws are obvious, and we describe them as follows.

(1) For Ali et al.'s protocol, $R_n^{\mathrm{new}} = r_i \oplus h(\mathrm{ID}_i^* \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, H(\mathrm{BIO}_i))$ should be added in step 2 of the password change phase, and R_n^{new} is also needed to replace the previous R_n in the smartcard.

(2) For Shuai et al.'s protocol, when performing step 2 of the password change phase, MD needs to additionally compute $D_i^{\mathrm{new}} = a_i \oplus h(\mathrm{ID}_i \,||\, \mathrm{PW}_i^{\mathrm{new}} \,||\, R_i)$ and replaces D_i with D_i^{new} in MD.

5. Conclusion

In the past few years, many three-factor authentication protocols have been proposed for WMSN and the similar environment. But, most of them are vulnerable to some inherent security defects more or less. In this paper, we briefly review and cryptanalyze the two quite recent and typical authentication protocols with key agreement presented by Ali et al. and Shuai et al., respectively. Firstly, we point out that although Ali et al. tried to overcome the security defects in the previous scheme and provide security proof with BAN logic and simulation under AVISPA, they are still vulnerable to offline dictionary guessing attack, desynchronization attack, and privileged insider attack and even contain a serious design flaw in the password change phase. Secondly, we demonstrate that Shuai et al.'s protocol is also insecure against offline dictionary guessing attack and privileged insider attack and has a design flaw in the password change phase. Thereafter, we put forward some possible countermeasures to eliminate these security weaknesses. Note that in this paper, the assumption that an attacker can simultaneously obtain both the secret information on the smartcard (mobile device) and the biometrics of the user is a trivial case, but it still cannot be ignored since security is one of the most important factors to consider in designing a protocol. Otherwise, if it is not based on this assumption, the attacker will require higher time complexity when carrying out offline ID and password dictionary attacks on the two protocols. Our efforts highlight that it is important to be aware of potential security risks in designing authentication protocols for WMSN and the similar kinds. This also indicates the necessity of our work.

Data Availability

(1) The reference data [19] used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s12652-018-1015-9]). (2) The reference data [24] used to support the findings of this study have been deposited in the [Hindawi] repository ([DOI: 10.1155/2019/8145087]).

Conflicts of Interest

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

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Project no. 61672007) and Science and Technology Innovation Guidance Project 2017 (Project no. 201704030605).

References

[1] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, "On the security and privacy of internet of things architectures and systems," in *Proceedings of the 2015 International Workshop on Secure Internet of Things (SIoT)*, pp. 49–57, IEEE, Vienna, Austria, September 2015.

[2] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Systems Journal*, pp. 1–12, 2019.

[3] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "Iot healthcare analytics: the importance of anomaly detection," in *Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 994–997, IEEE, Crans-Montana, Switzerland, May 2016.

[4] S. H. Shah, A. Iqbal, and S. S. A. Shah, "Remote health monitoring through an integration of wireless sensor networks, mobile phones & cloud computing technologies," in *Proceedings of the 2013 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 401–405, IEEE, San Jose, CA, USA, October 2013.

[5] G. Mois, T. Sanislav, and S. C. Folea, "A cyber-physical system for environmental monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 6, pp. 1463–1471, 2016.

[6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, ACM, Atlanta, GA, USA, September 2002.

[7] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.

[8] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the internet of things," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147–159, 2011.

[9] J. Spins, "Security protocols for sensor networks," *Wireless Networks*, vol. 5, pp. 521–534, 2002.

[10] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-sap: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[11] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[12] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and anonymous two-factor authentication protocol for healthcare applications with wireless medical sensor networks," *Multimedia Systems*, vol. 23, no. 2, pp. 195–205, 2017.

[13] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655, 2016.

[14] A. K. Das, A. K. Sutrala, V. Odelu, and A. Goswami, "A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1899–1933, 2017.

[15] J. Srinivas, D. Mishra, and S. Mukhopadhyay, "A mutual authentication framework for wireless medical sensor networks," *Journal of Medical Systems*, vol. 41, no. 5, p. 80, 2017.

[16] F. Wu, X. Li, A. K. Sangaiah et al., "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 82, pp. 727–737, 2018.

[17] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Generation Computer Systems*, vol. 80, pp. 483–495, 2018.

[18] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[19] R. Ali, A. K. Pal, S. Kumari, A. K. Sangaiah, X. Li, and F. Wu, "An enhanced three factor based authentication protocol using wireless medical sensor networks for healthcare monitoring," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–22, 2018.

[20] T. Hayajneh, B. Mohd, M. Imran, G. Almashaqbeh, and A. Vasilakos, "Secure authentication for remote patient monitoring with wireless medical sensor networks," *Sensors*, vol. 16, no. 4, p. 424, 2016.

[21] C.-H. Liu and Y.-F. Chung, "Secure user authentication scheme for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 59, pp. 250–261, 2017.

[22] S. Challa, A. K. Das, V. Odelu et al., "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 69, pp. 534–554, 2018.

[23] Y. Xie, S. Zhang, X. Li, Y. Li, and Y. Chai, "Cascp: efficient and secure certificateless authentication scheme for wireless body area networks with conditional privacy-preserving," *Security and Communication Networks*, vol. 2019, Article ID 5860286, 13 pages, 2019.

[24] M. Shuai, B. Liu, N. Yu, and L. Xiong, "Lightweight and secure three-factor authentication scheme for remote patient monitoring using on-body wireless networks," *Security and Communication Networks*, vol. 2019, Article ID 8145087, 14 pages, 2019.

[25] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2016.

[26] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[27] J. Mo, Z. Hu, H. Chen, and W. Shen, "An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 4520685, 12 pages, 2019.

[28] T. H. Kim, C. Kim, and I. Park, "Side channel analysis attacks using am demodulation on commercial smart cards with seed," *Journal of Systems and Software*, vol. 85, no. 12, pp. 2899–2908, 2012.

[29] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer Science & Business Media, Berlin, Germany, 2008.

[30] D. He, S. Zeadally, L. Wu, and H. Wang, "Analysis of handover authentication protocols for mobile wireless networks using identity-based public key cryptography," *Computer Networks*, vol. 128, pp. 154–163, 2017.

[31] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[32] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[33] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: an underestimated threat," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1242–1254, ACM, Vienna, Austria, October 2016.

[34] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[35] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 1–15, 2014.

[36] T.-H. Chen, Y.-C. Chen, W.-K. Shih, and H.-W. Wei, "An efficient anonymous authentication protocol for mobile pay-tv," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1131–1137, 2011.

[37] J. Wei, W. Liu, and X. Hu, "Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 77, no. 3, pp. 2255–2269, 2014.

[38] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-to-peer Networking and Applications*, vol. 9, no. 1, pp. 223–244, 2016.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8347213, 20 pages
https://doi.org/10.1155/2020/8347213

Research Article

Secure Outsourced Medical Data against Unexpected Leakage with Flexible Access Control in a Cloud Storage System

Xingguang Zhou,[1,2] Jianwei Liu,[1] Zongyang Zhang,[1] and Qianhong Wu[1]

[1]*School of Cyber Science and Technology, Beihang University, Beijing, China*
[2]*Civil Aviation Management Institute of China, Beijing, China*

Correspondence should be addressed to Zongyang Zhang; zongyangzhang@buaa.edu.cn

Received 30 August 2019; Revised 1 December 2019; Accepted 8 January 2020; Published 10 February 2020

Guest Editor: Kewei Sha

The application of cloud storage system has been deployed widely in recent years. A lot of electronic medical records (EMRs) are collected and uploaded to the cloud for scalable sharing among the authority users. It is necessary to guarantee the confidentiality of EMRs and the privacy of EMR owners. To achieve this target, we summarize a series of attack behaviors in the cloud storage system and present the security model against many types of unexpected privacy leakage. Privacy of unassailed EMRs is guaranteed in this model, and the influence of privacy leakage is controlled in a certain scope. We also propose a role-based access control scheme to achieve flexible access control on these private EMRs. One can access medical records only if his/her role satisfies the defined access policy, which implies a fine-grained access control. Theoretical and experimental analyses show the efficiency of our scheme in terms of computation and communication.

1. Introduction

Cloud communication has been envisioned as one of the most influential technologies in the medical field. Without being measured face-to-face, medical staff could give guidance to patients in a real-time way, which greatly improves the healthcare quality. For instance, a patient with heart disease history can deploy a medical sensor at home for the purpose of health monitoring. His health data are uploaded to cloud server and used by remote hospitals. The doctors in their duties could download his EMR and prepare the patient for treatment if needed. This method brings convenience for both patients and hospitals. However, once a patient's data are uploaded to the cloud storage system, they lose the physical control over the data and the cloud provider can obtain the access on it. Privacy threats experienced by users of Google Inc., Apple Inc., and Amazon Inc. [1] clearly indicate that cloud is intrinsically insecure from the users' point of view [2]. Most users would like to keep their personal information confidential to outsiders, let alone those patients whose EMRs include a lot of sensitive information. Data confidentiality is one of the important security concerns in the cloud storage system.

A common solution for data confidentiality is to encrypt them using public key encryption [3] before transmitting to the cloud system. We first generate two cryptographic keys: one is public and the other is secret. The public key is distributed by the data owner who is responsible for encryption. The secret key is private and assigned to each recipient in duty for decryption, such as the medical staff who is responsible for a patient.

While public key encryption ensures data confidentiality commendably, we admit that no matter what measures we take, unexpected privacy leakage sometimes happens. There are mainly three potential leakage risks in a typical cloud storage system: vulnerable medical devices, a semitrust cloud server, and the association among EMRs themselves. In the first type of risk, the data collected by medical devices will be firstly handled in a local but relatively open place (such as in a ward or on an ambulance), then it is uploaded to the cloud. The data are easily accessible for unauthorized persons in an emergency. In the second type of risk, since the cloud

provider is a semitrusted organization, it honestly follows the rules but does everything possible to spy on the stored files. Patients' information might be leaked by internal staff of the cloud provider. The third type of risk is due to EMRs' internal association for a patient and his family, i.e., a father's heart attack record may reflect a similar heart disease of his son. Leaking one record might infer unassailed ones. Besides the above three potential risks, we also need to consider that there are many malicious adversaries who keep trying to gain access to the cloud storage system, i.e., the communication among the devices (such as body sensors), the cloud and the EMR recipients. There are lots of ways to do this [4]. One way is to break the random-number generator (RNG) [5] and thus gain the randomness used for encryption. Another way is to break into the cloud part. For instance, Albrecht and Paterson [6] introduced a powerful attack that an adversary runs malicious JavaScript in a targeted browser and completely recovers HTTP session cookies or user credentials such as passwords.

All these unexpected events might cause parts of database corrupted and data exposed. The cloud storage system should be resilient in the case of security breach. In other words, once the privacy leakage happens, the leakage effect should be controllable, so that the confidentiality of the "unleaked" information is guaranteed. Therefore, controllability of privacy leakage is another important security concern in the cloud storage system.

Furthermore, we need to consider that the cloud storage system is usually associated with a multiparty communication environment, as Figure 1 describes. The health data are scalably shared among authorized users. From the recipients' point of view, the access control manner needs to be flexible enough to deal with the changes of users' roles and permissions [7].

1.1. Our Contributions. Based on the aforementioned security concerns in the cloud storage system, we propose a leakage controllable scheme to achieve data confidentiality with a flexible access manner. Since our scheme is based on a multiuser access policy, it quite matches the real world's scenario which includes many users in a hierarchical organization. Specific techniques are highlighted as follows.

Privacy Leakage Controllability. As unexpected privacy leakage always happens in various forms, we propose new security models, called role-based access control against unexpected leakage (RBAC-UL) to capture further leakage on the remaining "unleaked" EMRs. RBAC-UL security is achieved if confidentiality of unassailed EMRs is still guaranteed.

Flexible Access Control. We offer an efficient approach to support fine-grained access control for a hierarchical healthcare organization. A user can comprehend an EMR only if his identity satisfies the associated access policy.

Scalable Data Sharing. It is achieved by letting higher-level medical staff delegate access privilege for his subordinates.

Constant Size Ciphertext. Our scheme achieves constant size of encapsulated EMR no matter how many users satisfy the defined access policy.

Rigorous Security Analysis. To ensure that our proposal is qualified enough for the series of security concerns, we present rigorous security analysis. The analysis shows that our proposal achieves a high privacy-preserving capacity, where data confidentiality, leakage controllability, and access control flexibility can be achieved simultaneously in the cloud storage system.

2. Related Work

Privacy-preserving access control in the cloud storage system has received more and more attention recently. Cryptography and authentication methods are utilized in the cloud network to offer secure healthcare services via wireless communications [8]. For the security of EMR, encryption is an efficient and cost-saving choice to guarantee patients' privacy. A lot of prominent schemes have been proposed to achieve this target. The scheme applying identity-based encryption (IBE) [9, 10] presents efficient solutions for the body sensor network. While considering fine-grained sharing of encrypted data, attribute-based encryption (ABE) [11, 12] is promising. This is because ABE provides differential access privileges for a set of users such as healthcare providers and allows flexibility in designating the access privileges of individual users over the encrypted data [13]. An immediate attribute modification method is used to achieve fine-grained user revocation and the outsourced e-health records security [14]. The searchable ABE scheme is a promising technique that can ensure the protection of patients' private information without compromising on performance [15]. When applying ABE schemes in the medical systems, it shows the security and flexibility as the user tries to access the outsourced EMRs. Besides, a role-based access control framework [16, 17] is proposed by using hierarchical identity-based broadcast encryption (HIBBE) [18] without ABE, which ensures the security, scalability, and flexibility for the outsourced EMRs. A secure role-based cloud storage system for encrypted patient-centric health records is achieved in commercial healthcare systems [19]. An auditing revocable privacy-preserving access control scheme for the e-health records shows the efficiency of RBAC in terms of communication and computation [20]. An enhancing medical data security scheme in the cloud using RBAC provides security to the data over an alien environment [21].

Although the aforementioned schemes devote to securing the outsourced EMRs, they are unable to deal with the situation of unexpected privacy leakage, let alone to minimize its effect. In a cloud storage system, the leakage threats mainly include secret credential leakage [22, 23], encapsulation-related randomness leakage [24, 25], internal files, accounts or other records leakage, etc. The target of our paper is to minimize the impact of leakage in the event that these unexpected issues happen. We notice that a lot of schemes have been put forward theoretically against these unexpected leakages, including the public-key encryption

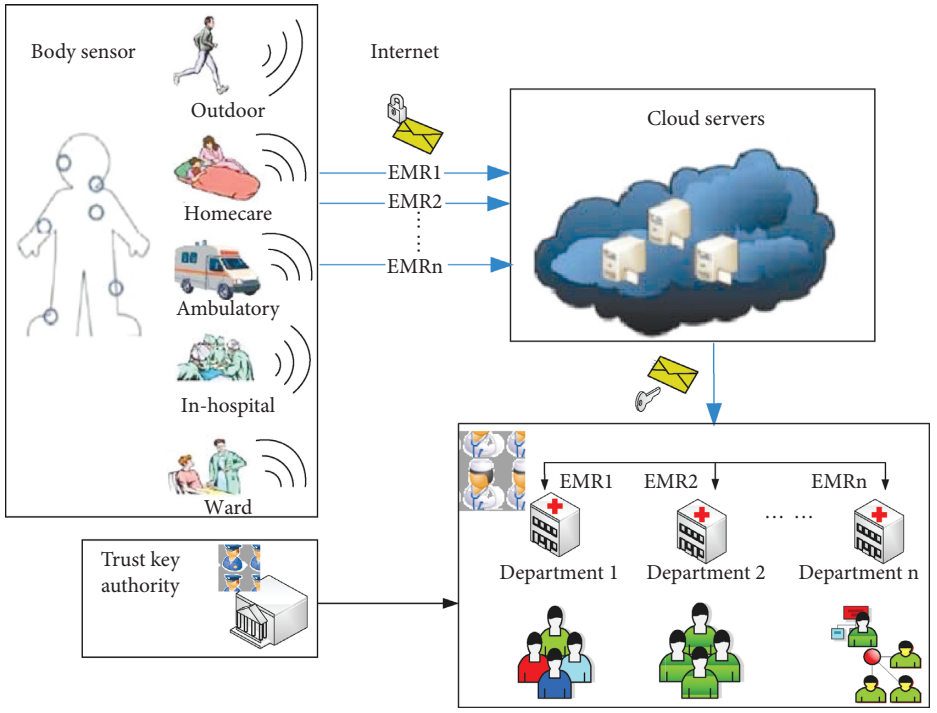


Figure 1: A medical cloud communication environment.

[26–28] and the identity-based encryption schemes [29–31]. They are different from our RBAC-UL mechanism. The former mainly guarantee the confidentiality of the remaining "unleaked" records, while ours not only ensures the confidentiality of "unleaked" data, but also achieves scalable sharing and flexible access control of all the outsourced EMRs.

3. Preliminaries

This section provides some mathematical basis for our proposal. The notations that are used in our scheme are given in Table 1. For ease of description, some of them are borrowed from [16, 17].

3.1. Bilinear Groups. Let $\mathcal{G}$ be a group generation algorithm that takes a security parameter λ as input and outputs the description of a bilinear group $(N, \mathbb{G}, \mathbb{G}_T, e)$. In our case, $\mathcal{G}$ outputs $(N = p_1p_2p_3p_4, G, G_T, e)$ where p_1, p_2, p_3, p_4 are distinct prime factors, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3p_4$, and e: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ is an efficient bilinear map satisfying the two properties: (i) bilinearity: For all $g, h \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$; (ii) non-degeneracy: there exists at least a generator g in $\mathbb{G}$ such that $e(g, g)$ generates $\mathbb{G}_T$. We, respectively, denote the subgroups of order p_1, p_2, p_3, p_4 in $\mathbb{G}$ by $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$, and $\mathbb{G}_{p_4}$. We use $G_{p_ip_j}$ $(1 \le i, j \le 4)$ to denote the subgroup of order p_ip_j in $\mathbb{G}$. These four subgroups additionally satisfy the orthogonality property, i.e., $\forall h_i \in \mathbb{G}_{p_i}$ and $h_j \in \mathbb{G}_{p_j}$ for $i \ne j$, $e(h_j, h_j) = 1$. This orthogonality property will be a principal tool in our constructions. Composite-order bilinear groups were first introduced in [32].

Table 1: Notations.

Notation	Description
λ	Security parameter
R	Atom role for medical staff
$\overrightarrow{R}$	Role for medical staff
$S_{\overrightarrow{R}}$	Atom role set for $\overrightarrow{R}$
P	Access policy
S_P	Atom role set for P
MSK	Master secret key
EMR	Electronic medical record
PPT	Probabilistic polynomial time
$\mathrm{SC}_{\overrightarrow{R}}$	Secret credential for a role $\overrightarrow{R}$
En	The encapsulated EMR
$\mathrm{Pref}(\overrightarrow{R})$	Prefix of $\overrightarrow{R}$, defined as $\{(R_1, \ldots, R_{d'}): d' \le d\}$
$\mathrm{Pref}(P)$	Prefix of P, defined as $\cup_{\overrightarrow{R} \in P} \mathrm{Pref}(\overrightarrow{R})$

3.2. Theoretical Assumptions. Our security analysis is based on the following mathematical assumptions.

Assumption 1. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, e) \xleftarrow{R} \mathcal{G}, g_1 \xleftarrow{R} G_{p_1}, g_3 \xleftarrow{R} G_{p_3}, g_4 \xleftarrow{R} G_{p_4},$$
$$D = (\mathbb{G}, g_1, g_3, g_4). \tag{1}$$

Assumption 1 determines whether a given element T is randomly chosen from G or from $G_{p_1p_3p_4}$, namely, $T \xleftarrow{R} G$ or $T \xleftarrow{R} G_{p_1p_3p_4}$. The advantage of an algorithm $\mathcal{A}$ that outputs a bit $b \in \{0, 1\}$ in breaking Assumption 1 is defined as

$$\mathrm{Adv1}_A(\lambda) = \left|\Pr\left[A(D, T \xleftarrow{R} G) = 1\right] - \Pr\left[A\left(D, T \xleftarrow{R} G_{p_1p_3p_4}\right) = 1\right]\right| - \frac{1}{2}. \tag{2}$$

Definition 1. $\mathcal{G}$ satisfies Assumption 1 if $\mathrm{Adv1}_{\mathcal{A}}(\lambda)$ is a negligible function for any PPT algorithm $\mathcal{A}$.

Assumption 2. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\begin{aligned}\mathbb{G} &= \left(N = p_1p_2p_3p_4, G, G_T, e\right) \xleftarrow{R} \mathcal{G}, g_1 \xleftarrow{R} G_{p_1}, \\ &\quad g_3 \xleftarrow{R} G_{p_3}, g_4 \xleftarrow{R} G_{p_4}, \\ D &= \left(\mathbb{G}, g_1, g_3, g_4\right).\end{aligned} \tag{3}$$

Assumption 2 determines whether a given element is $T \xleftarrow{R} G_{p_1p_2p_4}$ or $T \xleftarrow{R} G_{p_1p_4}$. The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in breaking the Assumption 2 is defined as

$$\mathrm{Adv2}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_2p_4}\right) = 1\right] - \Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_4}\right) = 1\right]\right| - \frac{1}{2}. \tag{4}$$

Definition 2. $\mathcal{G}$ satisfies the Assumption 2 if $\mathrm{Adv2}_{\mathcal{A}}(\lambda)$ is a negligible function for any PPT algorithm $\mathcal{A}$.

Assumption 3. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\begin{aligned}\mathbb{G} &= \left(N = p_1p_2p_3p_4, G, G_T, e\right) \xleftarrow{R} \mathcal{G}, g_1 \xleftarrow{R} G_{p_1}, g_3 \xleftarrow{R} G_{p_3}, \\ &\quad g_4 \xleftarrow{R} G_{p_4}, D_{23} \xleftarrow{R} G_{p_2p_3}, A_{12} \xleftarrow{R} G_{p_1p_2}, \\ D &= \left(\mathbb{G}, g_1, g_3, g_4, D_{23}, A_{12}\right).\end{aligned} \tag{5}$$

Assumption 3 determines whether a given element is $T \xleftarrow{R} G_{p_1p_2p_3}$ or $T \xleftarrow{R} G_{p_1p_3}$. The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in breaking Assumption 3 is defined as

$$\mathrm{Adv3}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_2p_3}\right) = 1\right] - \Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_3}\right) = 1\right]\right| - \frac{1}{2}. \tag{6}$$

Definition 3. $\mathcal{G}$ satisfies Assumption 3 if $\mathrm{Adv3}_{\mathcal{A}}(\lambda)$ is a negligible function for any PPT algorithm $\mathcal{A}$.

Assumption 4. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\begin{aligned}\mathbb{G} &= \left(N = p_1p_2p_3p_4, G, G_T, e\right) \xleftarrow{R} \mathcal{G}, g_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3}, \\ &\quad g_4 \xleftarrow{R} G_{p_4}, W_{14} \xleftarrow{R} G_{p_1p_4}, E_{12} \xleftarrow{R} G_{p_1p_2}, \\ D &= \left(\mathbb{G}, g_2, g_3, g_4, W_{14}, E_{12}\right).\end{aligned} \tag{7}$$

Assumption 4 determines whether a given element is $T \xleftarrow{R} G_{p_2p_4}$ or $T \xleftarrow{R} G_{p_1p_2p_4}$. The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in breaking Assumption 4 is defined as

$$\mathrm{Adv4}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_2p_4}\right) = 1\right] - \Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_2p_4}\right) = 1\right]\right| - \frac{1}{2}. \tag{8}$$

Definition 4. $\mathcal{G}$ satisfies Assumption 4 if $\mathrm{Adv4}_{\mathcal{A}}(\lambda)$ is a negligible function for any PPT algorithm $\mathcal{A}$.

Assumption 5. Given a group generator $\mathcal{G}$, we define the following distribution:

$$\begin{aligned}\mathbb{G} &= \left(N = p_1p_2p_3p_4, G, G_T, e\right) \xleftarrow{R} \mathcal{G}, g_1 \xleftarrow{R} G_{p_1}, \\ &\quad g_4 \xleftarrow{R} G_{p_4}, D_{23} \xleftarrow{R} G_{p_2p_3}, \\ D &= \left(\mathbb{G}, g_1, g_4, D_{23}\right).\end{aligned} \tag{9}$$

Assumption 5 determines whether a given element is $T \xleftarrow{R} G$ or $T \xleftarrow{R} G_{p_1p_2p_4}$. The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in breaking Assumption 5 is defined as

$$\mathrm{Adv5}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\mathcal{A}(D, T \xleftarrow{R} G) = 1\right] - \Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_2p_4}\right) = 1\right]\right| - \frac{1}{2}. \tag{10}$$

Definition 5. $\mathcal{G}$ satisfies Assumption 5 if $\mathrm{Adv5}_{\mathcal{A}}(\lambda)$ is a negligible function for any PPT algorithm $\mathcal{A}$.

4. System Model

In Figure 1, we describe a typical medical cloud storage system. It is a multiuser setting environment consisting of four entities: an EMR owner, a cloud server, an EMR recipient, and a trusted key authority (TKA).

The EMR owner is usually a patient who is monitored by different types of medical sensors. His/her medical records are sent via wireless networks. Each record is encrypted and associated with its own access policy and then stored on the cloud server for sharing with the entitled medical staff.

The cloud provides a large number of servers for many organizations. It is honest but curious, i.e., it obeys rules of the cloud system, but could do everything possible to spy on the stored EMRs.

The EMR recipient consists of groups of medical staff who are entitled to read the patients' EMRs and provide services for them. The medical staff are semitrusted. If they are authorized, they do not reveal any information. Otherwise, they might be potential adversaries. The staff at the higher-level is responsible for managing lower-level ones, which derives a tree-like organization. For example, the role of a nurse consisting of ordered atom roles "department psychiatry, chief doctor, head nurse, nurse," is administrated by the head nurse whose role is "department psychiatry, chief doctor, head nurse." The head nurse is administrated by the chief doctor and so on. We group the chief doctor, the head nurse, and the nurse in one access policy, where all of them are responsible for a certain patient. The patient is identified by his name or identity information. Each medical staff can encapsulate the patient's EMR, but only the one whose role satisfies the corresponding access policy can decapsulate it.

The TKA is responsible for generating and distributing system parameters.

A role-based access control scheme consists of the following algorithms:

$(\mathrm{PK}, \mathrm{MSK}) \longleftarrow \mathrm{Setup}(\lambda, n)$: the setup algorithm is run by TKA. It takes as inputs a security parameter λ and a maximal size n of users. It outputs a masker key MSK and a public key PK.

$\mathrm{SC}^{\overrightarrow{R}} \longleftarrow \mathrm{SCGen}(\mathrm{PK}, \mathrm{MSK}, \overrightarrow{R})$: the secret credential generation algorithm of medical staff takes as inputs a public key PK, a master key MSK, and a role $\overrightarrow{R}$ for a medical staff. It outputs a secret credential $\mathrm{SC}^{\overrightarrow{R}}$ for the medical staff with role $\overrightarrow{R}$.

$\mathrm{SC}^{\overrightarrow{R}} \longleftarrow \mathrm{SCDeleg}(\mathrm{PK}, \mathrm{SC}^{\overrightarrow{R'}}, R)$: the secret credential delegation algorithm of medical staff takes as inputs a public key PK, a secret credential $\mathrm{SC}^{\overrightarrow{R'}}$ for a medical staff with role $\overrightarrow{R'}$, and an atom role R. It returns the secret credential $\mathrm{SC}^{\overrightarrow{R}}$ for the medical staff with role $\overrightarrow{R} = (\overrightarrow{R'}, R)$. The medical staff with role $\overrightarrow{R'}$ is the supervisor of the one with role $\overrightarrow{R}$.

$\mathrm{En} \longleftarrow \mathrm{EMREnc}(\mathrm{PK}, P, \mathrm{EMR})$: the EMR encapsulation algorithm takes as inputs a public key PK, an access policy P, and an EMR file EMR. It outputs an encapsulated EMR file En.

$\mathrm{EMR} \longleftarrow \mathrm{EMRDec}(\mathrm{PK}, \overrightarrow{R}, \mathrm{En}, \mathrm{SC}^{\overrightarrow{R}})$: the EMR decapsulation algorithm takes as inputs a public key PK, a medical staff's role R, an encapsulated EMR En, and a secret credential $\mathrm{SC}^{\overrightarrow{R}}$ for the medical staff with role $\overrightarrow{R}$. It outputs an EMR file EMR.

5. Security Requirements

In practice, all entities except TKA are likely to attack the cloud storage system. A dishonest party may try to get useful information from the encapsulated EMRs, which is not authorized to access or derive from the leaked EMRs. In the context of such attack, our scheme is expected to meet the following security requirements.

(i) Data privacy: EMRs need to be obfuscated before being uploaded and securely stored on cloud servers, until an authorized user downloads and deobfuscates them.

(ii) Leakage controllability: when a privacy leakage happens unavoidably, it must be possible to minimize the leakage effect, which means the privacy of nonleaked EMRs should be guaranteed.

5.1. The Adversary Model for RBAC-UL. The adversary model for RBAC-UL is aimed to satisfy the requirement of *leakage controllability*. Since the content of EMRs may be internally related, partial leakage might expose information on those EMRs. Therefore, it is necessary to clarify what it means for the unleaked EMRs to remain confidential. In the RBAC-UL model, we assume two roles: an adversary and a simulator. The adversary's goal is to collect as much information from the unleaked EMRs as possible, i.e., the corrupted EMRs, the encapsulated EMRs, and the randomness used for encapsulation. The simulator acts like a normal person with neutral characters: he can get the same input as the adversary when a leakage happens, and he has the ability to corrupt the EMR owners to learn their EMRs. Apart from that, the simulator cannot get any further information. We claim that if any adversary cannot obtain more information from the unleaked EMRs than the simulator, then the security of the remaining EMRs is guaranteed.

We formally define the adversary model for RBAC-UL, which is inspired by the work [29]. In the model, the corrputed EMR is encapsulated with a target access policy set P^* containing all the medical staff who are allowed to decapsulate. The adversary is allowed to do the following: (a) it can obtain secret credentials associated with roles $R \notin \mathrm{Pref}(P^*)$, which implies the adversary can collude any medical staff with roles that do not satisfy the target access policy; (b) it can obtain all the targeted encapsulated EMRs, $\overrightarrow{\mathrm{En}} = \{\mathrm{En}_1, \mathrm{En}_2, \ldots, \mathrm{En}_n\}$; (c) it can randomly corrupt any encapsulated EMRs from $\overrightarrow{\mathrm{En}}$ and then obtain their files $\{\mathrm{EMR}_i\}_{i\in[1,n]}$ with the randomness used for encapsulation $\{r_i\}_{i\in[1,n]}$, which implies that the adversary has the ability to corrupt the EMR owners.

We use two security games for further illustration. The first one is played between an adversary $\mathscr{A}$ and a challenger $\mathscr{C}$. It describes what an adversary could obtain in the real world. The second one is played between a simulator $\mathscr{S}$ and a challenger $\mathscr{C}$. It describes what a simulator could obtain in an ideal experiment.

The real game $\mathrm{Game}_{\mathrm{real},\mathscr{A}}(\lambda)$ of RBAC-UL:

Setup: the challenger $\mathscr{C}$ runs Setup to obtain the system parameter PK and gives PK to the adversary $\mathscr{A}$.

Challenge: $\mathscr{A}$ outputs a set of EMRs, $\overrightarrow{\mathrm{EMR}} = \{\mathrm{EMR}_1, \mathrm{EMR}_2, \ldots, \mathrm{EMR}_n\}$ on which it wishes to challenge, together with a set of access policy $P^* = \{P_1^*, P_2^*, \ldots, P_n^*\}$ including all the broadcast groups that it wishes to attack. In our case, a broadcast group represents a group of medical staff who are eligible to read a certain

kind of EMR. For example, the medical staff in the access policy group P_2^* are authorized to fetch EMR_2, but no authority for other EMRs. Each access policy P_i^* should satisfy that for all the secret credential queries issued in Query Phase 1, there is $\overrightarrow{R} \notin \mathrm{Pref}(P_i^*)$. $\mathscr{C}$ randomly chooses $r[i] \longleftarrow Z_p$ for each EMR_i, where $p = |\mathbb{G}|$ and $i \in \{1, 2, \ldots, n\}$, and computes $\mathrm{En}_i \longleftarrow \mathrm{Enc}(\mathrm{PK}, P_i^*, \mathrm{EMR}_i, r_i)$. Finally, the challenger $\mathscr{C}$ returns the encapsulated EMRs $\overrightarrow{\mathrm{En}} = \{\mathrm{En}_1, \mathrm{En}_2, \ldots, \mathrm{En}_n\}$ to $\mathscr{A}$.

Query Phase 1: $\mathscr{A}$ issues a secret credential query for a medical staff associated with role $\overrightarrow{R}$. The challenger $\mathscr{C}$ generates a secret credential for $\overrightarrow{R}$ and returns it to $\mathscr{A}$.

Challenge: $\mathscr{A}$ outputs a set of EMRs, $\overrightarrow{\mathrm{EMR}} = \{\mathrm{EMR}_1, \mathrm{EMR}_2, \ldots, \mathrm{EMR}_n\}$ on which it wishes to challenge, together with a set of access policy $P^* = \{P_1^*, P_2^*, \ldots, P_n^*\}$ including all the broadcast groups that it wishes to attack. In our case, a broadcast group represents a group of medical staff who are eligible to read a certain kind of EMR. For example, the medical staff in the access policy group P_2^* are authorized to fetch EMR_2, but no authority for other EMRs. Each access policy P_i^* should satisfy that for all the secret credential queries issued in Query Phase 1, there is $\overrightarrow{R} \notin \mathrm{Pref}(P_i^*)$. $\mathscr{C}$ randomly chooses $r[i] \longleftarrow Z_p$ for each EMR_i, where $p = |\mathbb{G}|$ and $i \in \{1, 2, \ldots, n\}$, and computes $\mathrm{En}_i \longleftarrow \mathrm{Enc}(\mathrm{PK}, P_i^*, \mathrm{EMR}_i, r_i)$. Finally, the challenger $\mathscr{C}$ returns the encapsulated EMRs $\overrightarrow{\mathrm{En}} = \{\mathrm{En}_1, \mathrm{En}_2, \ldots, \mathrm{En}_n\}$ to $\mathscr{A}$.

Corrupt: the adversary $\mathscr{A}$ outputs a set $I \subseteq [1, n]$ to $\mathscr{C}$ on which $\mathscr{A}$ wishes to corrupt. $\mathscr{C}$ corrupts the corresponding EMRs to get $(\mathrm{EMR}_i, r_i)_{i \in I}$ and returns them to the adversary.

Query Phase 2: $\mathscr{A}$ issues a secret credential query for the medical staff with role $\overrightarrow{R}$ such that $\overrightarrow{R} \notin \mathrm{Pref}(P^*)$. The challenger responds the same as Query Phase 1.

Output: the adversary $\mathscr{A}$ outputs a bit $\mathrm{Out}_{\mathscr{A}}$.

The simulated game $\mathrm{Game}_{\mathrm{sim},\mathscr{S}}(\lambda)$ of RBAC-UL:

Setup: the simulator $\mathscr{S}$ gets system parameters from the challenger $\mathscr{C}$.

Challenge: $\mathscr{S}$ outputs a set of EMRs, $\overrightarrow{\mathrm{EMR}} = \{\mathrm{EMR}_1, \mathrm{EMR}_2, \ldots, \mathrm{EMR}_n\}$ on which it wishes to challenge, together with a set of access policy $P^* = \{P_1^*, P_2^*, \ldots, P_n^*\}$ including all the broadcast groups that it wishes to attack. The challenger $\mathscr{C}$ gets these inputs, but gives no feedback to the adversary $\mathscr{A}$.

Corrupt: $\mathscr{S}$ outputs a set $I \subseteq [1, n]$ to $\mathscr{C}$. The challenger $\mathscr{C}$ picks up the corresponding $\{\mathrm{EMR}_i\}_{i \in I}$ and returns them to the simulator.

Output: the simulator $\mathscr{S}$ outputs a bit $\mathrm{Out}_{\mathscr{S}}$.

We claim that if for every PPT adversary there exists a PPT simulator who can generate an indistinguishable output without seeing any encapsulated EMR and randomness, then the scheme achieves RBAC-UL security.

Definition 6. The advantage of a RBAC-UL adversary $\mathscr{A}$ against a RBAC scheme Γ with a simulator $\mathscr{S}$ is defined as follows:

$$\mathrm{Adv}_{\Gamma,\mathscr{S},\mathscr{A}}^{RBAC-UL}(\mathscr{A}) = \left|\Pr\left[\mathrm{Game}_{\mathrm{real},\mathscr{A}}(\lambda) = 1\right] - \Pr\left[\mathrm{Game}_{\mathrm{sim},\mathscr{S}}(\lambda) = 1\right]\right|. \tag{11}$$

Definition 7 (RBAC-UL Security). Given an RBAC scheme, if no PPT distinguisher $\mathscr{D}$ can distinguish the output from $\mathrm{Game}_{\mathrm{real},\mathscr{A}}$ and $\mathrm{Game}_{\mathrm{sim},\mathscr{S}}$, namely,

$$\left|\Pr[\mathscr{D}(\mathrm{Out}_{\mathscr{A}} = 1)] - \Pr[\mathscr{D}(\mathrm{Out}_{\mathscr{S}} = 1)]\right| = \epsilon, \tag{12}$$

where ϵ is a negligible function in the security parameter, then we claim that the scheme achieves RBAC-UL security.

5.2. The Adversary Model for RBAC-IND. The adversary model for RBAC-Indistinguishability (RBAC-IND) is aimed to satisfy the requirement of *data privacy*. The indistinguishability can be illustrated as follows: a recipient (the medical staff who are eligible to get access on one EMR) generates a credential pair; a sender (EMR owner) encapsulates one out of two EMRs and send it to the adversary; the RBAC-IND adversary tries to find out which one it was. Here, importantly, the adversary has the authority to issue secret credential query. If the adversary cannot distinguish an encapsulation of challenge EMR from an encapsulation of random message, we claim that our system achieves RBAC-IND security.

The RBAC-IND model is defined by a security game played between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$. We apply the full security notion [33, 34] to the RBAC-IND model. That means the adversary can adaptively output the access policy that it wishes to attack during the system interaction.

The security game $\mathrm{Game}_{\Gamma,\mathscr{A}}^{\mathrm{RBAC-IND}}(\lambda)$: let $\Gamma = (\mathrm{Setup}, \mathrm{SCGen}, \mathrm{SCDeleg}, \mathrm{EMREnc}, \mathrm{EMRDec})$ be a role-based access control scheme.

Setup: the challenger runs the setup algorithm to obtain a public key PK and gives it to the adversary $\mathscr{A}$.

Query Phase 1: $\mathscr{A}$ issues a secret credential query for the medical staff with role $\overrightarrow{R}$. The challenger generates a secret credential for $\overrightarrow{R}$ and gives it to the adversary.

Challenge: the adversary $\mathscr{A}$ outputs two equal-length EMR files, EMR_0, and EMR_1 on which it wishes to challenge. $\mathscr{A}$ outputs a challenge access policy P^* either. The access policy P^* should satisfy that for all the secret credential queries for $\overrightarrow{R}$ issued in Query Phase, $\overrightarrow{R} \notin \mathrm{Pref}(P^*)$. The challenger flips a random coin $\beta \in \{0, 1\}$ and encapsulates EMR_β under the challenge access policy P^*. Then, it returns the encapsulated EMR to $\mathscr{A}$.

Query Phase 2: $\mathcal{A}$ issues a credential query for the medical staff with role $\overrightarrow{R}$ such that $\overrightarrow{R} \notin \text{Pref}(P^*)$. The challenger responds the same as in Query Phase.

Guess: the adversary $\mathcal{A}$ guesses $\beta' \in \{0, 1\}$. It ouputs 1 if $\beta = \beta'$ and 0 otherwise. We say $\mathcal{A}$ succeeds if $\beta = \beta'$.

Definition 8. Given an RBAC scheme Γ, we define the probability advantage for an RBAC-IND adversary $\mathcal{A}$ of winning the security game $\text{Game}_{\Gamma,\mathcal{A}}^{RBAC-IND}(\lambda)$ to be

$$\text{Adv}_{\Gamma,\mathcal{A}}^{\text{RBAC-IND}}(\lambda) = \left|2 \cdot \Pr\left[\text{Game}_{\Gamma,\mathcal{A}}^{\text{RBAC-IND}}(\lambda) = 1\right] - 1\right|. \tag{13}$$

Definition 9 (RBAC-IND security). Given an RBAC scheme Γ, if for each polynomial-time RBAC-IND adversary, its advantage $\text{Adv}_{\Gamma,\mathcal{A}}^{\text{RBAC-IND}}(\lambda)$ is a negligible function in security parameter λ, then the scheme Γ achieves RBAC-IND security.

6. Proposed Solution

Our technical solution leverages hierarchical identity-based encryption (HIBE) [35] and extends it to a multireceiver scenario. In HIBE, an encryptor can only encrypt a single path, which implies either repetitive encryption or constrained access policy for multiple receivers. Our role-based access control solution supports fine-grained access by encapsulating EMRs to any subset of hierarchically organized users, which is based on HIBBE. Furthermore, it resists many types of unexpected leakage, so that the leakage effect can be controlled in a certain scope. The following subsections show how we achieve this target. In Section 6.1, we construct a one-bit RBAC scheme with one-sided public leakage (1SPL) functionality. 1SPL means there exists a public procedure that given the one-bit encapsulation message En^1 of "1" can compute the randomness r under which the encapsulation applied to "1" would generate En^1. En^1 is used to denote the encapsulation message of "1." The idea comes from the notion of one-side public openability [36]. In Section 6.2, we provide security analysis for the proposed one-bit RBAC scheme with 1SPL. In Section 6.4, we provide a reduction showing that if a one-bit RBAC scheme with 1SPL functionality is secure, the normal multibit scheme with RBAC-UL model is secure. Our solution achieves data privacy, leakage controllability, and flexible access control.

6.1. Construction of One-Bit RBAC Scheme with 1SPL

Setup (λ, n). The system setup algorithm is run by TKA. It chooses a bilinear group G of order N, and random elements g_1, g_2, g_3, g_4 from $G_{p_1}, G_{p_2}, G_{p_3}, G_{p_4}$, random exponents $u_{11}, u_{12}, \ldots, u_{1n}, u_4, x_1, x_4, \omega_4 \xleftarrow{R} Z_N$, and computes $U_{1i} \xleftarrow{R} g_1^{u_{1i}}$, $U_4 \xleftarrow{R} g_4^{u_4}$, $X_1 \xleftarrow{R} g_1^{x_1}$, $X_4 \xleftarrow{R} g_4^{x_4}$, $W_4 \xleftarrow{R} g_4^{\omega_4}$, $U_{1i,4} \xleftarrow{R} U_{1i}U_4$, $W_{14} \xleftarrow{R} g_1 W_4$, $X_{14} \xleftarrow{R} X_1 X_4$ for $i \in [1, n]$. It outputs a public key $\text{PK} = \left\{N, \{U_{1i,4}\}_{i\in[1,n]}, X_{14}, W_{14}, g_4\right\}$ and a master secret key $\text{MSK} = \left\{g_1, g_3, \{U_{1i}\}_{i\in[1,n]}, X_1\right\}$.

SCGen (PK, MSK, $\overrightarrow{R}$). This is the secret credential generation algorithm. For the medical staff with role $\overrightarrow{R} = (R_1, \ldots, R_d)$, we denote $I = \left\{i\text{: } R_i \in S_{\overrightarrow{R}}\right\}$. When a medical staff at the top-level joins a hospital organization, TKA generates a secret credential $\text{SC}^{\overrightarrow{R}}$ for them:

$$\begin{aligned} K_1 &= g_1^r g_3^{r_3}, \\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r \cdot g_3^{r_3'}, \\ E_j &= \left\{U_{1j}^r \cdot g_3^{r_j}\right\}_{j\in[i,n]/I}, \end{aligned} \tag{14}$$

where $r, r_3, r_3', \{r_j\}_{j\in([1,n]/I)} \xleftarrow{R} Z_N$.

SCDeleg (PK, $\text{SC}^{\overrightarrow{R'}}$, R). This is the secret credential delegation algorithm. A junior medical staff with role $\overrightarrow{R} = (\overrightarrow{R'}, R)$ is authenticated by a supervisor with role $\overrightarrow{R'}$. His supervisor delegates a secret credential for them:

$$\begin{aligned} K_1 &= K_1' g_1^{r'} g_3^{\widetilde{r_3}}, \\ K_2 &= K_2' \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^{r'} \cdot (E_i')_{i\in(I/I')}^R \cdot g_3^{\widetilde{r_3'}}, \\ E_j &= \left\{E_j' \cdot U_{1,j}^{r'} \cdot g_3^{r_j'}\right\}_{j\in[1,n]/I} = \left\{U_{1,j}^{r+r'} \cdot g_3^{r_j'+r_j}\right\}_{j\in[1,n]/I}, \end{aligned} \tag{15}$$

where $I' = \left\{i\text{: } R_i \in S_{\overrightarrow{R'}}\right\}$ and $r', \widetilde{r_3}, \widetilde{r_3'}, \{r_j'\}_{j\in[i,n]/I} \xleftarrow{R} Z_N$. It can be computed as

$$\begin{aligned} K_1 &= g_1^{\widehat{r}} g_3^{\widehat{r_3}}, \\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^{\widehat{r}} \cdot g_3^{\widehat{r_3'}}, \\ E_j &= \left\{U_{1j}^{\widehat{r}} \cdot g_3^{\widehat{r_j}}\right\}_{j\in[i,n]/I}. \end{aligned} \tag{16}$$

The delegated credential is well formed as if it is generated by TKA with SCGen algorithm.

EMREnc (PK, P, EMR): this is the EMR encapsulation algorithm. For an access policy P, we denote $\text{II} = \{i\text{: } R_i \in S_P\}$. When a single bit 0 from EMR data needs to be encapsulated under P, a medical staff chooses random $s, t_4, t_4' \xleftarrow{R} Z_N$ and computes $\text{En}_1 = (\prod_{i\in\text{II}} U_{1i,4}^{R_i} \cdot X_{14})^s g_4^{t_4}$, $\text{En}_2 = W_{14} g_4^{t_4}$. When a

single bit $\underline{1}$ needs to be encapsulated, a medical staff sets $\mathrm{En}_1, \mathrm{En}_2 \xleftarrow{R} G$.

EMRDec$(\mathrm{PK}, \overrightarrow{R}, \mathrm{En}, \mathrm{SC}^{\overrightarrow{R}})$: this is the EMR decapsulation algorithm. The medical staff with role satisfied by an access policy P can use his secret credential to recover all one-bit messages for EMR data. If $e(\mathrm{En}_1, K_1) = e(\mathrm{En}_2, K_2)$, a medical staff returns bit 0. Otherwise, he returns bit 1.

Correctness: We need to verify when input a well-formed encapsulation $\mathrm{En} = (\mathrm{En}_1, \mathrm{En}_2)$ with a valid credential $\mathrm{SC}^{\overrightarrow{R}}$ for 0 bit, whether $e(\mathrm{En}_1, K_1) = e(\mathrm{En}_2, K_2)$ holds.

$$e(\mathrm{En}_1, K_1) = e\left(\left(\prod_j U_{1j,4}^{id_j} \cdot X_{14}\right)^s g_4^{t_4}, g_1^r g_3^r\right),$$
$$e(\mathrm{En}_2, K_2) = e\left(g_1^s \cdot g_4^{\omega_4 s + t_4'}, \left(g_1^{u_{11} \cdot id_1} \cdots g_1^{u_{1j \cdot id_j}} \cdot g_1^{x_1}\right)^r \cdot g_3^{r_3'}\right). \tag{17}$$

Due to the orthogonality property, we get $e(\mathrm{En}_1, K_1) = e(\mathrm{En}_2, K_2) = e(g_1^s, g_1^{r \cdot (\sum_{i=1}^{j} u_{1i} \cdot id_1 + x_1)})$. Therefore, when $\mathrm{En} = (\mathrm{En}_1, \mathrm{En}_2)$ is a well-formed EMR encapsulation, the decapsulation algorithm can correctly recover EMR with a valid credential $\mathrm{SC}^{\overrightarrow{R}}$.

6.2. Security Analysis of One-Bit RBAC Scheme. We prove the security by contradiction. Assume a PPT adversary can break the one-bit RBAC scheme in polynomial time. Then we solve a series of hard-to-solve problems based on subgroup decision assumptions, which are introduced in Section 3.2. Since no PPT algorithm could solve these problems, we reach a contradiction and conclude our proposed scheme is secure.

Theorem 1. *Suppose $\mathbb{G}$ is a group of composite order $N = p_1 p_2 p_3 p_4$, equipped with an efficient bilinear map. Suppose that the Assumption 1–5 hold in $\mathbb{G}$. Then our one-bit RBAC scheme is secure under the formal security model.*

We apply the dual system encapsulation technique [37] to the one-bit RBAC scheme, where the encapsulated message En and the credential SC can take one of two indistinguishable form: normal form and semifunctional form. The correlation between them is shown in Table 2. "√" means is decapsulation allowed and "×" means decapsulation is not allowed. When all the EMR encapsulations and credentials are semifunctional, the adversary obtains no information for the challenge encapsulated EMR since none of the given credential is useful to decapsulate it.

In the next section, we show that no PPT algorithm can distinguish between $\mathrm{Game}_{\mathrm{real}}$ and $\mathrm{Game}_{\mathrm{final}}$. All the components in the encapsulation of $\mathrm{Game}_{\mathrm{final}}$ are random elements, so it does not leak any EMR information. The indistinguishability between those games proves Theorem 1.

Semifunctional ciphertext: a user runs EHRGen to construct a normal ciphertext $(\mathrm{En}_1', \mathrm{En}_1')$. Then they pick up random exponents $x, z_c \in Z_N$ and sets $\mathrm{En}_1 = \mathrm{En}_1' g_2^{xz_c}$, $\mathrm{En}_2 = \mathrm{En}_2' g_2^{x}$.

Table 2: Correlation of normal and semifunctional forms.

	Normal En	Semifunctional En
Normal SC	√	√
Semifunctional SC	√	×

Semifunctional secret credential TKA runs the algorithm SCGenM to generate a normal key $(K_1', K_2', \{E_j\}_{j \in [1,n]/I})$. It chooses random exponents $\gamma, z_k, \{z_j\}_{j \in [1,n]/I} \in Z_N$. The semifunctional key is set as $K_1 = K_1' g_2^{\gamma}, K_2 = K_2' g_2^{\gamma \cdot z_k}, \{E_j = E_j^A g_2^{\gamma z_j}\}$.

It is straightforward that the EHRDecM algorithm correctly outputs EHR when decrypting a semifunctional ciphertext by a semifunctional key since the added elements in G_{p_2} can be cleared due to orthogonality property. However, the blinding factor is multiplied by an additional term $e(g_2, g_2)^{\gamma x (z_k + \sum_{i \in (\mathbb{I}/I)} z_i R_i - z_c)}$. If $z_c = z_k + \sum_{i \in (\mathbb{I}/I)} z_i R_i$, decryption still works. In this case, we call the secret credential is nominally semifunctional. We prove Theorem 1 through following games between an adversary and a challenger.

$\mathrm{Game}_{\mathrm{real}}$: this is the real game.

$\mathrm{Game}_{\mathrm{real}'}$: this game is the same as $\mathrm{Game}_{\mathrm{real}}$ except that all the secret credential queries are answered by the secret credential generation algorithm, not by the secret credential delegation algorithm.

$\mathrm{Game}_{\mathrm{res}}$: this game is the same as $\mathrm{Game}_{\mathrm{real}'}$ except that the adversary cannot ask for secret credentials for the roles which are the prefixes of the challenge role modulo p_2. Namely, it is not allowed that, for any queried role $\overrightarrow{R} = (R_1, \ldots, R_d)$, $\exists \overrightarrow{R^*} = (R_1^*, R_2^*, \ldots, R_{d'}^*) \in \mathrm{Pref}(P^*)$ with $d' \le d$, s.t. $\forall i \in [1, d'], R_i = R_i^* \bmod p_2$. P^* is the set of challenge access policy.

Game_k: this game is identical with $\mathrm{Game}_{\mathrm{res}}$ except that the EMR encapsulation given to adversary is semifunctional and the first k credentials are semifunctional $(0 \le k \le q)$ for bit 0. We notice that in Game_q, the EMR encapsulation and all credentials are semifunctional.

$\mathrm{Game}_{\mathrm{final}'}$: this game is identical with Game_q except that challenge EMR encapsulation is a semifunctional encapsulation for a random message in subgroup of $\mathbb{G}$ for bit 0, not one of the messages given by the adversary.

$\mathrm{Game}_{\mathrm{final}}$: this game is identical with $\mathrm{Game}_{\mathrm{final}'}$ except that $\mathrm{Game}_{\mathrm{final}}$ replaces the challenge EMR encapsulation of 0 by a pair of random points in the full group $\mathbb{G}$.

6.3. Proof of Theorem 1. In this section, we use six lemmas to prove Theorem 1. Each lemma demonstrates the indistinguishability between the neighbouring games.

Lemma 1. *For any PPT algorithm $\mathcal{A}$, it holds that: $Game_{real} Adv_{\mathcal{A}}(\lambda) = Game_{real'} Adv_{\mathcal{A}}(\lambda)$.*

Proof of Lemma 1. We note that the secret credentials are identically distributed whether they are generated by the credential generated algorithm or by the credential delegation algorithm. So, there is no difference between $\text{Game}_{\text{real}}\text{Adv}_{\mathscr{A}}$ and $\text{Game}_{\text{real}'}\text{Adv}_{\mathscr{A}}$ from the adversary's view. □

Lemma 2. *Suppose there exists a PPT algorithm $\mathscr{A}$ such that $|Game_{real'}Adv_{\mathscr{A}}(\lambda) - Game_{res}Adv_{\mathscr{A}}(\lambda)| = \epsilon_1$. Then, we can build a polynomial-time algorithm $\mathscr{B}$ with advantage $\epsilon_1/3$ in breaking Assumption 1.*

Proof of Lemma 2. If there exists a PPT adversary $\mathscr{A}$ that distinguishes $\text{Game}_{\text{real}}$ and Game_{res} with probability ϵ_1, by the definition of Game_{res}, $\mathscr{A}$ knows that it issues a secret credential query for the medical staff with role $\overrightarrow{R} = \{R_1, \ldots, R_d\}$ from others satisfying that $\exists R^* = (R_1^*, R_2^*, \ldots, R_{d'}^*) \in \text{Pref}(P^*)$ with $d' \le d$, s.t. $\forall i \in [1, d']$, $R_i = R_i^* \bmod p_2$. Then the factor of N can be extracted by computing $\gcd(R_i - R_i^*, N)$, from which we design an algorithm $\mathscr{B}$ breaking Assumption 1 as follows.

$\mathscr{B}$ receives g_1, g_3, g_4, and produces a nontrivial factor of N by computing $r = \gcd(R_i - R_i^*, N)$. We want to use r to generate a point in $\mathbb{G}(Q)$, where $\mathbb{G}(Q)$ denotes the unique subgroup with order $\prod_{j\in Q} p_j$ ($Q \subseteq [4]$). We set $2 \in Q$ but $k \notin Q$ so that $\mathbb{G}(Q)$ can be used to test T with orthogonality. Enumerate the cases for r, N/r, and the resulting k and Q in Table 3. As N/r is the complementary set of r, it covers all the possibilities for the subgroup with different orders.

Due to the rule that $2 \in Q$ but $k \notin Q$, we get from Table 3 at least one choice of k that allows us to use r or N/r to construct a point in $\mathbb{G}(Q)$. $\mathscr{B}$ immediately decides T by orthogonality. For example, if r is p_1, T is chosen from G_{p_1} or $G_{p_1p_2}$. Also we can get elements from $\mathbb{G}(Q)$, i.e., we select $X \xleftarrow{R} G_{p_2p_3p_4}$. Then $\mathscr{B}$ learns whether T has a G_{p_2} component or not by testing if $e(T, X) = 1$. If not, T has a G_{p_2} component. From the point of $\mathscr{A}$'s view, the choice of i is independent, and $\mathscr{B}$ at least has 1/3 chance to pick an i that works.

Compared with Game_{res}, the challenge encapsulation of a 0 bit is replaced with a semifunctional one in Game_0, meaning its components are multiplied by points in G_{p_2}. As the adversary does not know the factor of $N = p_1p_2p_3p_4$, it cannot determine whether the components of the challenge encapsulation of a 0 are in $G_{p_1p_4}$ or in $G_{p_1p_2p_4}$. Hence, the adversary is not able to know which form the given challenge encapsulation is. □

Lemma 3. *Suppose there exists a PPT algorithm $\mathscr{A}$ such that $|Game_{res}Adv_{\mathscr{A}}(\lambda) - Game_0Adv_{\mathscr{A}}(\lambda)| = \epsilon_2$. We can build a polynomial-time algorithm $\mathscr{B}$ with advantage ϵ_2 in breaking Assumption 2.*

Table 3: Possibilities for $r, N/r, k, Q$.

r	N/r	k	Q
p_1	$p_2p_3p_4$	1	2 3 4
p_2	$p_1p_3p_4$	1 3 4	2
p_3	$p_1p_2p_4$	3	1 2 4
p_4	$p_1p_2p_3$	4	1 2 3
p_1p_2	p_3p_4	3 4	1 2
p_1p_3	p_2p_4	1 3	2 4
p_1p_4	p_2p_3	1 4	2 3

Proof of Lemma 3. The input of algorithm $\mathscr{B}$ is the challenge tuple (g_1, g_3, g_4, T) of Assumption 2. $\mathscr{B}$ has to decide whether T is in $G_{p_1p_4}$ or in $G_{p_1p_2p_4}$.

Setup: on inputs (g_1, g_3, g_4, T), $\mathscr{B}$ picks random exponents $\{a_i\}_{i\in[1,n]}, b, x, y, \omega$ from Z_N, and sets $U_{1i} \xleftarrow{R} g_1^{a_i}$, $X_1 \xleftarrow{R} g_1^b, X_4 \xleftarrow{R} g_4^x, U_4 \xleftarrow{R} g_4^y$, $W_4 \xleftarrow{R} g_4^\omega$, $U_{1i,4} \xleftarrow{R} U_{1i}U_4, X_{14} \xleftarrow{R} X_1X_4$, $W_{14} \xleftarrow{R} g_1W_4$. It sends public paramter $\{N, U_{1i,4}, X_{14}, W_{14}, g_4\}_{i\in[1,n]}$ to $\mathscr{A}$

Query Phase 1: when the adversary $\mathscr{A}$ issues a secret credential query for a medical staff with role $\overrightarrow{R} = (R_1, \ldots, R_d)$, $\mathscr{B}$ randomly chooses exponents $r, r_3, r_3', \{r_j\}_{j\in([1,n]/I)} \xleftarrow{R} Z_N$ where $I = \left\{i\colon R_i \in S_{\overrightarrow{R}}\right\}$ and sets

$$K_1 = g_1^r g_3^{r_3},$$
$$K_2 = \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r \cdot g_3^{r_3'}, \tag{18}$$
$$E_j = \left\{U_{1j}^r \cdot g_3^{r_j}\right\}_{j\in[i,n]/I}.$$

It has the same distribution as that of the normal secret credential.

Challenge: $\mathscr{A}$ outputs two EMR files EMR_0 and EMR_1 and a challenge access policy P^*. The challenge access policy P^* must satisfy the property that no revealed role in Query Phase 1 was a prefix of its components. $\mathscr{B}$ picks a random coin $\beta \xleftarrow{R} \{0, 1\}$ and gives the challenge EMR encapsulation as following. We denote that $\text{II}^* = \{i\colon R_i^* \in S_{P^*}\}$.

(i) $\text{EMR}_\beta = 0$. $\mathscr{B}$ lets $z_c = \sum_{i\in\text{II}^*} a_iR_i^* + b$ and sets $\text{En}_1 \xleftarrow{R} T^{z_c}$, $\text{En}_2 \xleftarrow{R} T$. If $\mathscr{B}$'s challenge bit $\beta = 0$, $T \xleftarrow{R} G_{p_1p_2p_4}$. We write $T = (g_1W_4)^s g_4^{t_4'} g_2^{x_2}$ for random $s, t_4', x_2 \xleftarrow{R} Z_N$ and get

$$\text{En}_1 = \left((g_1W_4)^s g_4^{t_4'} g_2^{x_2}\right)^{z_c} = \left(\prod_{i\in\text{II}^*} U_{1i,4}^{R_i} \cdot X_{14}\right)^s g_4^{\left(s\omega + t_4'\right)\cdot z_c - \left(y\sum_{i\in\text{II}^*} id_i + x\right)s} \cdot g_2^{x_2\cdot z_c},$$
$$\text{En}_2 = (g_1W_4)^s g_4^{t_4'} g_2^{x_2} = W_{14}^s g_4^{t_4'} g_2^{x_2}. \tag{19}$$

This implicitly sets $t_4 = (s\omega + t_4') \cdot z_c - (y\sum_{i\in\mathbb{I}^*} R_i + x)s \bmod p_4$ and $x_2' = x_2 \cdot z_c \bmod p_2$. The challenge encapsulation is semifunctional formed in $\text{Game}_0^{\mathscr{A}}$. If $\mathscr{B}$'s challenge bit $\beta = 1$, $T \xleftarrow{R} G_{p_1p_4}$. We write $T = (g_1W_4)^s g_4^{t_4'}$ for random $s, t_4' \xleftarrow{R} Z_N$, and get

$$\begin{aligned}\text{En}_1 &= \left((g_1W_4)^s g_4^{t_4'}\right)^{Z_c} = \left(\prod_{i\in\text{II}^*} U_{1i,4}^{R_i} \cdot X_{14}\right)^s \\ &\quad g_4^{(s\omega+t_4')\cdot z_c - \left(y\sum_{i\in\text{II}^*} id_i + x\right)s}, \\ \text{En}_2 &= (g_1W_4)^s g_4^{t_4'} = W_{14}^s g_4^{t_4'}.\end{aligned} \tag{20}$$

This implicitly sets $t_4 = (s\omega + t_4') \cdot z_c - (y\sum_{i\in\text{II}^*} R_i + x)s \bmod p_4$. The challenge EMR encapsulation is normally formed in Game_{res}.

(ii) $\text{EMR}_\beta = 1$. $\mathscr{B}$ sets $\text{En}_1, \text{En}_2 \xleftarrow{R} \mathbb{G}$.

Query Phase 2: Query Phase 1 is repeated adaptively except that $\overrightarrow{R} \notin \text{Pref}(P^*)$.

Guess: the adversary $\mathscr{A}$ outputs a guess that it is in Game_0 or Game_{res}. The simulator $\mathscr{B}$ guesses $T \xleftarrow{R} G_{p_1p_4}$ if $\mathscr{A}$ decides it is in Game_{res} ($\beta = 1$). $\mathscr{B}$ outputs $T \xleftarrow{R} G_{p_1p_2p_4}$ if $\mathscr{A}$ decides it is in Game_0 ($\beta = 0$). If $\mathscr{A}$ has the advantage ε_2 to distinguish Game_{res} and Game_0, $\mathscr{B}$ can break Assumption 2 with advantage ϵ_2.

Game_{k-1} and Game_k are two distinguishable games. The way to decide whether the kth queried credential is normal or semifunctional is to decide whether the credential components are in $G_{p_1p_3}$ or in $G_{p_1p_2p_3}$. This is computationally difficult without knowing factor $N = p_1p_2p_3p_4$. □

Lemma 4. *Suppose there exists a PPT algorithm $\mathscr{A}$ such that $|\text{Game}_{k-1}Adv_{\mathscr{A}}(\lambda) - \text{Game}_kAdv_{\mathscr{A}}(\lambda)| = \varepsilon_3$. Then, we can build a polynomial-time algorithm $\mathscr{B}$ with advantage ε_3 in breaking Assumption 3.*

Proof of Lemma 4. The input of $\mathscr{B}$ is the challenge tuple $(g_1, g_3, g_4, D_{23}, A_{12}, T)$ of Assumption 3. $\mathscr{B}$ has to decide whether T is in $G_{p_1p_3}$ or in $G_{p_1p_2p_3}$.

Setup: $\mathscr{B}$ receives $g_1, g_3, g_4, D_{23}, A_{12}, T$. It picks random exponents $\{a_1\}_{i\in[1,n]}, b, x, y, \omega$ from Z_N, and sets $U_{1i} \xleftarrow{R} g_1^{a_i}$, $X_1 \xleftarrow{R} g_1^b$, $X_4 \xleftarrow{R} g_4^x$, $U_4 \xleftarrow{R} g_4^y$, $W_4 \xleftarrow{R} g_4^\omega$, $U_{1i,4} \xleftarrow{R} U_{1i}U_4$, $X_{14} \xleftarrow{R} X_1X_4$, $W_{14} \xleftarrow{R} g_1$ W_4 and $k \xleftarrow{R} [0, q]$. It sends the public parameters $\left\{N, U_{1i,4}, X_{14}, W_{14}, g_4\right\}_{i\in[1,n]}$ to $\mathscr{A}$.

Query Phase 1: when $\mathscr{A}$ requests the ℓth credential for $\overrightarrow{R} = (R_1, \ldots, R_d)$ where $I = \left\{i\colon R_i \in S_{\overrightarrow{R}}\right\}$, we consider three cases: $\ell < k$, $\ell > k$, and $\ell = k$.

(i) When $\ell < k$, $\mathscr{B}$ creates a semifunction credential by picking up random exponents $r, z, z^A, \left\{z_j\right\}_{j\in([1,n]/I)}$ from Z_N and setting

$$\begin{aligned}K_1 &= g_1^r D_{23}^z, \\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r \cdot D_{23}^{z'}, \\ E_j &= \left\{U_{1,j}^r D_{23}^{z_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{21}$$

Consider D_{23} as $g_2^{y_2}g_3^{y_3}$ for random $y_2, y_3 \xleftarrow{R} Z_N$, then

$$\begin{aligned}K_1 &= g_1^r g_2^{y_2z} g_3^{y_3z}, \\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r \cdot g_2^{z'y_2} g_3^{z'y_3}, \\ E_j &= \left\{U_{1,j}^r g_3^{y_3\cdot z_j} g_2^{y_2\cdot z_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{22}$$

This is a properly distributed semifunctional credential.

(ii) When $\ell > k$, $\mathscr{B}$ creates a normal credential by invoking the usual credential generation algorithm.

(iii) When $\ell = k$, $\mathscr{B}$ creates the kth credential. $\mathscr{B}$ lets $z_k = \sum_{i\in I} a_iR_i + b$ and sets

$$\begin{aligned}K_1 &\xleftarrow{R} T, \\ K_2 &\xleftarrow{R} T^{z_k}, \\ E_j &\xleftarrow{R} \left\{T^{a_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{23}$$

If $T \xleftarrow{R} G_{p_1p_3}$, $T = g_1^r g_3^{r_3}$ for random $r, r_3 \xleftarrow{R} Z_N$, then

$$\begin{aligned}K_1 &= g_1^r g_3^{r_3}, \\ K_2 &= \left(g_1^r g_3^{r_3}\right)^{z_k} = \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r g_3^{r_3\cdot z_k}, \\ E_j &= \left\{\left(g_1^r g_3^{r_3}\right)^{a_j}\right\}_{j\in([1,n]/I)} = \left\{U_{1,j}^r g_3^{r_3\cdot a_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{24}$$

It has the same distribution as the normal credential.

If $T \xleftarrow{R}_R G_{p_1p_2p_3}$, we write $T = g_1^r g_3^{r_3} g_2^{t_2}$ for random $r, r_3, t_2 \xleftarrow{R} Z_N$, then

$$\begin{aligned}K_1 &= g_1^r g_3^{r_3} g_2^{t_2}, \\ K_2 &= \left(g_1^r g_3^{r_3} g_2^{t_2}\right)^{Z_k} = \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r g_3^{r_3\cdot z_k} g_2^{t_2\cdot z_k}, \\ E_j &= \left\{\left(g_1^r g_3^{r_3} g_2^{t_2}\right)^{a_j}\right\} = \left\{U_{1,j}^r g_3^{r_3\cdot a_j} g_2^{t_2\cdot a_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{25}$$

It has the same distribution as the semifunctional credential.

Challenge: at some points, $\mathscr{A}$ decides that it obtains enough secret credentials, it outputs two EMR files EMR_0 and EMR_1 and a challenge access policy P^*.

This policy must satisfy that no revealed role in Query Phase 1 was a prefix of its components. $\mathcal{B}$ picks up a random coin $\mathcal{B} \xleftarrow{R} \{0,1\}$ and gives the challenge EMR encapsulation as follows:

(i) $\mathrm{EMR}_{\beta_R} = 0$. $\mathcal{B}$ picks up $s_4, s_4' \xleftarrow{R} Z_N$ and sets $\mathrm{En}_1 \xleftarrow{R} A_{12}^{z_c} g_4^{s_4}, \mathrm{En}_2 \xleftarrow{R} A_{12} g_4^{s_4'}$. Consider A_{12} as $g_1^s g_2^{x_2}$ for random $s, x_2 \xleftarrow{R} Z_N$, and get

$$\begin{aligned}\mathrm{En}_1 &= \left(\prod_{i\in \mathrm{II}^*} U_{1i}^{R_i^*} \cdot X_{14}\right)^s \cdot g_2^{x_2 \cdot z_c} \cdot g_4^{s_4 - xs - ys\sum_{i\in \mathrm{II}^*} R_i^*},\\ \mathrm{En}_2 &= g_1^s g_2^{x_2} g_4^{s_4'} = \left(g_1 g_4^{\omega}\right)^s \cdot g_4^{s' - \omega s} \cdot g_2^{x_2}\\ &= g_1^s W_4^s g_4^{s_4' - \omega s} g_2^{x_2}\\ &= W_{14}^s g_4^{t_4'} g_2^{x_2}.\end{aligned} \tag{26}$$

This implicitly sets $t_4 = s_4 - xs - ys\sum_{i\in \mathrm{II}^*} R_i^* \bmod p_4$, $x_2' = x_2 \cdot z_c \bmod p_2$ and $t_4' = s\imath - \omega s \bmod p_4$ for $\mathbb{I}^* = \{i\colon R_i^* \in S_{P*}\}$.

(ii) $\mathrm{EMR}_\beta = 1$. $\mathcal{B}$ sets $\mathrm{En}_1, \mathrm{En}_2 \xleftarrow{R} \mathbb{G}$.

The challenge encapsulation for $\mathrm{EMR}_\beta = 0$ is formed as the semifunctional form with $z_c = \sum_{i\in \mathrm{II}^*} a_i R_i^* + b$. Since from $\mathrm{Game}_{\mathrm{res}}$, the role $\overrightarrow{R} = (R_1, \ldots, R_d)$ associated with the kth secret credential is not a prefix of the challenge receiver role $\overrightarrow{R}^* = (R_1^*, \ldots, R_d^*)$ modulo p_2, the variables z_c and z_k are randomly distributed to the adversary $\mathcal{A}$. The relationship between z_c and z_k do not help $\mathcal{A}$ to distinguish the two games.

Query Phase 2: Query Phase 1 is repeated except $\overrightarrow{R} \notin \mathrm{Pref}(P^*)$.

Guess: the adversary $\mathcal{A}$ outputs a guess that it is in Game_{k-1} or Game_k.

$\mathcal{B}$ outputs $T \xleftarrow{R} G_{p_1 p_3}$ if $\mathcal{A}$ decides it is in Game_{k-1}, where all components in the kth secret credential by algorithm $\mathcal{B}$ are in $G_{p_1 p_3}$. Otherwise, $\mathcal{B}$ outputs $T \xleftarrow{R} G_{p_1 p_2 p_3}$ if $\mathcal{A}$ decides it is in Game_k, where all components in the kth secret credential by algorithm $\mathcal{B}$ are in $G_{p_1 p_2 p_3}$. If $\mathcal{A}$ has the advantage ε_3 to distinguish Game_{k-1} and Game_k, $\mathcal{B}$ can break Assumption 3 with advantage ϵ_3. □

Lemma 5. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $|Game_q Adv_{\mathcal{A}}(\lambda) - Game_{final'} Adv_{\mathcal{A}}(\lambda)| = \varepsilon_4$. Then we can build a polynomial-time algorithm $\mathcal{B}$ with advantage ϵ_4 in breaking Assumption 4.*

Proof of Lemma 5. The input of algorithm $\mathcal{B}$ is the challenge tuple $(g_2, g_3, g_4, W_{14}, E_{12}, T)$ of Assumption 4. Algorithm $\mathcal{B}$ has to answer whether T is in $G_{p_2 p_4}$ or in $G_{p_1 p_2 p_4}$.

Setup: $\mathcal{B}$ first receives $g_2, g_3, g_4, W_{14}, E_{12}, T$. It then picks random exponents $\{a_i\}_{i\in[1,n]}, b$ from Z_N, and sets $U_{1i,4} \xleftarrow{R} W_{14}^{a_i}$, $X_{14} \xleftarrow{R} W_{14}^b$. It sends these public parameters $\{N, U_{1i,4}, X_{14}, W_{14}, g_4\}_{i\in[1,n]}$ to $\mathcal{A}$.

Query Phase 1: When $\mathcal{A}$ requests the secret credential for the medical staff with role $\overrightarrow{R} = (R_1, \ldots, R_d)$, $\mathcal{B}$ lets $z_k = \sum_{i\in I} a_i R_i + b$, randomly chooses exponents t, y, y', r_3, r_3', $\{r_j\}_{j\in([1,n]/I)}, \{y_j\}_{j\in([1,n]/I)} \xleftarrow{R} Z_N$ where $I = \left\{i\colon R_i \in S_{\overrightarrow{R}}\right\}$, and sets

$$\begin{aligned}K_1 &= E_{12} g_2^y g_3^{r_3},\\ K_2 &= E_{12}^{t\cdot z_k} g_2^{y'} g_3^{r_3'},\\ E_j &= \left\{E_{12}^{t\cdot a_j} g_3^{r_j} g_2^{y_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{27}$$

Consider $E_{12} = g_1^{e_1} g_2^{e_2}$ for random $e_1, e_2 \xleftarrow{R} Z_N$, and get

$$\begin{aligned}K_1 &= g_1^{te_1} g_2^{te_2 + y} g_3^{r_3},\\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^{te_1} \cdot g_2^{z_k te_2 + y'} \cdot g_3^{r_3'},\\ E_j &= \left\{U_{1,j}^{te_1} \cdot g_2^{te_2 \cdot a_j + y_j} \cdot g_3^{r_j}\right\}_{j\in([1,n]/I)}.\end{aligned} \tag{28}$$

This implicitly sets $r = te_1$, $t_2 = te_2 + y$, $t_2' = z_k te_2 + y'$, $U_{1i} = g_1^{a_i}$, $X_1 = g_1^b$, and $t_j = te_2 \cdot a_j + y_j$. The simulated key is distributed as the semifunctional credential.

Challenge: $\mathcal{A}$ outputs two EMR files EMR_0 and EMR_1, and a challenge access policy P^*. This policy must satisfy that no revealed role in Query Phase 1 was a prefix of its components. $\mathcal{B}$ picks a random coin $\beta \xleftarrow{R} \{0,1\}$ and gives the challenge EMR encapsulation as follows. We denote that $\mathrm{II}^* = \{i\colon R_i^* \in S_{P*}\}$.

(i) $\mathrm{EMR}_\beta = 0$. $\mathcal{B}$ randomly chooses $s, x_2' \xleftarrow{R} Z_N$ and sets

$$\mathrm{En}_1 \xleftarrow{R} \left(\prod_{i\in \mathrm{II}^*} U_{1i,4}^{R_i} \cdot X_{14}\right)^s T^{x_2'}, \mathrm{En}_2 \xleftarrow{R} W_{14}^s T. \tag{29}$$

(a) If $\mathcal{B}$'s challenge bit is $\beta = 1$, then $T \xleftarrow{R} G_{p_1 p_2 p_4}$. Hence, the challenge ciphertexts En_1 and En_2 are random in $G_{p_1 p_2 p_4}$ as in $\mathrm{Game}_{\mathrm{final}'}$.

(b) If $\mathcal{B}$'s challenge bit is $\beta = 0$, then $T \xleftarrow{R} G_{p_2 p_4}$. We write $T = g_4^{t_4'} g_2^{x_2}$ for random $t_4', x_2 \xleftarrow{R} Z_N$ and get

$$\mathrm{En}_1 = \left(\prod_{i\in \mathbb{I}^*} U_{1i,4}^{R_i} \cdot X_{14}\right)^s \cdot g_4^{t_4' s x_2'} g_2^{x_2 s x_2'}, \mathrm{En}_2 = W_{14}^s g_4^{t_4' s} g_2^{x_2 s}. \tag{30}$$

(c) This implicitly sets $t_4 = t_4' x_2' \bmod p_4$. The challenge encapsulation is formed as the semifunctional form in Game_q.

(ii) $\mathrm{EMR}_\beta = 1$. $\mathcal{B}$ sets $\mathrm{En}_1, \mathrm{En}_2 \xleftarrow{R} \mathbb{G}$.

Query Phase 2: Query Phase 1 is repeated adaptively except $\overrightarrow{R} \notin \mathrm{Pref}(P^*)$.

Guess: the adversary $\mathcal{A}$ outputs a guess that it is in Game_q or $\mathrm{Game}_{\mathrm{final}'}$.

The simulator $\mathcal{B}$ guesses $T \xleftarrow{R} G_{p_2 p_4}$ if $\mathcal{A}$ decides it is in Game_q ($\beta = 0$). Otherwise, $\mathcal{B}$ outputs $T \xleftarrow{R} G_{p_1 p_2 p_4}$ ($\beta = 1$). If $\mathcal{A}$ has an advantage ε_4 to distinguish Game_q and $\mathrm{Game}_{\mathrm{final}'}$, $\mathcal{B}$ breaks Assumption 4 with advantage ε_4. Since

all the credentials and EMR encapsulations are semifunctional in Game_q, $\mathcal{A}$ cannot get any information about the challenge EMR encapsulation due to none of the given credentials are useful to decapsulate it. Hence, $\mathcal{A}$ cannot find that the challenge EMR encapsulation has been replaced by a random component. This implies the indistinguishability between Game_q and $\text{Game}_{\text{final}\prime}$. □

Lemma 6. *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $|Game_{final\prime}Adv_{\mathcal{A}}(\lambda) - Game_{final}Adv_{\mathcal{A}}(\lambda)| = \varepsilon_5$. Then we construct a PPT algorithm $\mathcal{B}$ with advantage ϵ_5 in breaking Assumption 5.*

Proof of Lemma 6. The input of $\mathcal{B}$ is the challenge tuple (g_1, g_4, D_{23}, T) of Assumption 5. $\mathcal{B}$ has to answer T is in $G_{p_1p_2p_4}$ or in G.

Setup: $\mathcal{B}$ first receives g_1, g_4, D_{23}, T. It then picks up random exponents $\{a_i\}_{i\in[1,n]}, b$ from Z_N, and sets $U_{1i} \xleftarrow{R} g_1^{a_i}$, $X_1 \xleftarrow{R} g_1^b$, $X_4 \xleftarrow{R} g_4^x$, $U_4 \xleftarrow{R} g_4^y$, $W_4 \xleftarrow{R} g_4^{\omega}$, $U_{1i,4} \xleftarrow{R} U_{1i}U_4$, $X_{14} \xleftarrow{R} X_1X_4$, $W_{14} \xleftarrow{R} g_1W_4$. It sends the public parameters $\{N, U_{1i,4}, X_{14}, W_{14}, g_4\}_{i\in[1,n]}$ to $\mathcal{A}$.

Query Phase 1: when $\mathcal{A}$ requests a secret credential for a medical staff with role $\overrightarrow{R} = (R_1, \ldots, R_d)$, $\mathcal{B}$ lets $z_k = \sum_{i\in I} a_iR_i + b$, chooses exponents r, d, d', $\{d_j\}_{j\in([1,n]/I)} \xleftarrow{R} Z_N$ where $I = \{i: R_i \in S_{\overrightarrow{R}}\}$, and sets

$$\begin{aligned} K_1 &= g_1^r D_{23}^d, \\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r D_{23}^{d'}, \\ E_j &= \left\{U_{1j}^r D_{23}^{d_j}\right\}_{j\in([1,n]/I)}. \end{aligned} \tag{31}$$

We write $D_{23} = g_2^{e_2}g_3^{e_3}$ for random $e_2, e_3 \xleftarrow{R} Z_N$ and get

$$\begin{aligned} K_1 &= g_1^r g_2^{e_2 d} g_3^{e_3 d}, \\ K_2 &= \left(\prod_{i\in I} U_{1i}^{R_i} \cdot X_1\right)^r g_2^{e_2 d'} \cdot g_3^{e_3 d'}, \\ E_j &= \left\{U_{1j}^r g_3^{e_3 d_j} \cdot g_2^{e_2 d_j}\right\}_{j\in i\in([1,n]/I)}. \end{aligned} \tag{32}$$

This implicitly sets $r_3 = e_3 d$, $t_2 = e_2 d$, $r_3' = e_3 d\prime$, $t_2' = e_2 d\prime$, $r_j = e_3 d_j$, and $t_j = e_2 d_j$. The simulated credential is distributed as the semifunctional credential.

Challenge: $\mathcal{A}$ outputs two EMR files EMR_0 and EMR_1, and a challenge access policy P^*. This policy must satisfy that no revealed role in Query Phase 1 was a prefix of its components. $\mathcal{B}$ picks up a random exponent z and a random coin $\beta \xleftarrow{R} \{0,1\}$. It gives the challenge encapsulation: $\text{En}_1 \xleftarrow{R} T, \text{En}_2 \xleftarrow{R} T^z$. If $\mathcal{B}$'s challenge bit is $\beta = 0$ then $T \xleftarrow{R} G$. Hence, the challenge ciphertexts En_1 and En_2 are random components in $\mathbb{G}$ as in $\text{Game}_{\text{final}}$.

If $\mathcal{B}$'s challenge bit is $\beta = 1$, then $T \xleftarrow{R} G_{p_1p_2p_4}$. Hence, the challenge ciphertexts En_1 and En_2 are random components in $G_{p_1p_2p_4}$ as in $\text{Game}_{\text{final}\prime}$.

Query Phase 2: repeat Query phase 1 except $\overrightarrow{R} \notin \text{Pref}(P^*)$.

Guess: the adversary $\mathcal{A}$ outputs a guess whether it is in $\text{Game}_{\text{final}\prime}$ or in $\text{Game}_{\text{final}}$. The simulator $\mathcal{B}$ guesses $T \xleftarrow{R} G_{p_1p_2p_4}$ if $\mathcal{A}$ decides it is in $\text{Game}_{\text{final}\prime}$ ($\beta = 1$). Otherwise, $\mathcal{B}$ outputs $T \xleftarrow{R} G$ ($\beta = 0$). If $\mathcal{A}$ has the advantage ε_5 to distinguish $\text{Game}_{\text{final}\prime}$ and $\text{Game}_{\text{final}}$, $\mathcal{B}$ can break Assumption 5 with advantage ε_5. $\text{Game}_{\text{final}}$ replaces the challenge encapsulation of 0 by a pair of random points in the full group. From the view of adversary, it cannot find that the challenge EMR encapsulation has been replaced by a random component in the full group or in the subgroup. Hence, it implies the indistinguishability between $\text{Game}_{\text{final}\prime}$ and $\text{Game}_{\text{final}}$. □

Proof of Theorem 1. If a group generator algorithm $\mathbb{G}$ satisfies Assumption i with advantage $\varepsilon_i'(1 \le i \le 5)$, then Lemmas 0–5 show that there is no polynomial time adversary to distinguish $\text{Game}_{\text{real}}$ and $\text{Game}_{\text{final}}$ with advantage $|\text{Game}_{\text{real}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{final}}\text{Adv}_{\mathcal{A}}(\lambda)|$, which can be expanded as follows:

$$\begin{aligned} &\left|\text{Game}_{\text{real}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{final}}\text{Adv}_{\mathcal{A}}(\lambda)\right| \\ &\le \left|\text{Game}_{\text{real}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{real}\prime}\text{Adv}_{\mathcal{A}}(\lambda)\right| + \left|\text{Game}_{\text{real}\prime}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{res}}\text{Adv}_{\mathcal{A}}(\lambda)\right| \\ &\quad + \left|\text{Game}_{\text{res}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_0\text{Adv}_{\mathcal{A}}(\lambda)\right| + \left|\text{Game}_0\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_1\text{Adv}_{\mathcal{A}}(\lambda)\right| \\ &\quad + \left|\text{Game}_1\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_2\text{Adv}_{\mathcal{A}}(\lambda)\right| + \cdots + \left|\text{Game}_{q-1}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_q\text{Adv}_{\mathcal{A}}(\lambda)\right| \\ &\quad \cdot \left|\text{Game}_q\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{final}\prime}\text{Adv}_{\mathcal{A}}(\lambda)\right| + \left|\text{Game}_{\text{final}\prime}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{final}}\text{Adv}_{\mathcal{A}}(\lambda)\right| \\ &= 3\varepsilon_1' + \varepsilon_2' + \varepsilon_3' + \varepsilon_4' + \varepsilon_5'. \end{aligned} \tag{33}$$

All the components in $\text{Game}_{\text{final}}$ are random elements in $\mathbb{G}$, and the messages are hidden from the adversary. Therefore, if the group $\mathbb{G}$ with composite order $N = p_1p_2p_3p_4$ satisfies Assumption 1–5 with advantage $\varepsilon_1', \varepsilon_2', \varepsilon_3', \varepsilon_4', \varepsilon_5'$ respectively, then our one-bit RBAC is secure with advantage $3\varepsilon_1' + \varepsilon_2' + \varepsilon_3' + \varepsilon_4' + \varepsilon_5'$. □

6.4. From One-Bit RBAC with 1SPL to Multibit RBAC-UL. We provide security analysis for the RBAC-UL model. The key point is to reduce RBAC-UL security from a secured one-bit RBAC with 1SPL functionality. We use a specific 1SPL algorithm "LeakToOne" which exposes the randomness as if it is randomly chosen for bit 1, and fails with probability δ when it cannot find out the randomness to 1. In the security analysis, we assume that all the roles in the access policy set or its subset are ordered from high-level staff to the lower level one.

Theorem 2. *Let Γ be a one-bit RBAC scheme, and Γ^ℓ be the ℓ-bit RBAC scheme built from it. Let k be the number of leaked EMRs and δ be the failing probability of LeakToOne. Suppose there exists an RBAC-UL adversary $\mathcal{A}$, RBAC-UL simulator $\mathcal{S}$, and RBAC adversary $\mathcal{B}$. If Γ is secure with $Adv^{RBAC-IND}_{\Gamma,\mathcal{B}}(\lambda)$, then Γ^ℓ is secure with $Adv^{RBAC-UL}_{\Gamma^\ell,\mathcal{S},\mathcal{A}}(\lambda) \le k\ell \cdot Adv^{RBAC-IND}_{\Gamma,\mathcal{B}}(\lambda) + k\ell \cdot \delta$.*

We prove it by a series of game transitions.

(i) $\text{Game}^{\mathcal{A}}_{\text{real}}$: this is the real game.

(ii) $\text{Game}^{\mathcal{A}}_{\text{real1}}$: this game reselects the randomness for the "0" bit encapsulation at Corrupt phase.

(iii) $\text{Game}^{\mathcal{A}}_{\text{real2}}$: this game runs LeakToOne algorithm for the "0" bit encapsulation at Corrupt phase. If LeakToOne fails, it reselects the randomness as $\text{Game}^{\mathcal{A}}_{\text{real1}}$ does.

(iv) $\text{Game}^{\mathcal{A}}_{\text{real3}}$: compared with $\text{Game}^{\mathcal{A}}_{\text{real2}}$, it does nothing if LeakToOne algorithm fails.

(v) $\text{Game}^{\mathcal{A}}_{v}$: the first v bits from the challenge EMRs are replaced by bit "0" and then encapsulated. The remaining bits are encapsulated normally. At Corrupt phase, if it needs to open an encapsulation component to a "0" bit, it directly gives $\mathcal{A}$ the randomness it used when creating the encapsulation. If it needs to open an encapsulation to a "1" bit, it runs LeakToOne to find the randomness.

(vi) $\text{Game}^{\mathcal{A}}_{k\ell}$: all the bits from challenge EMRs are replaced by "0" and all the "0" bits are encapsulated.

(vii) $\text{Game}^{\mathcal{S}}_{\text{sim}}$: this game is run by the simulator $\mathcal{S}$.

In the next subsection, we show that no PPT algorithm can distinguish between $\text{Game}^{\mathcal{A}}_{\text{real}}$ and $Game^{\mathcal{A}}_{\text{real1}}$ and between $\text{Game}^{\mathcal{A}}_{\text{real1}}$ and $\text{Game}^{\mathcal{A}}_{\text{real2}}$. Then we demonstrate that, if any execution of LeakToOne fails at most δ, no PPT algorithm has advantage $k\ell\delta$ to distinguish between $\text{Game}^{\mathcal{A}}_{\text{real2}}$ and $\text{Game}^{\mathcal{A}}_{\text{real3}}$. Following that, if no adversary $\mathcal{B}$ has the advantage $\text{Adv}^{\text{RBAC–IND}}_{\Gamma,\mathcal{B}}(\lambda)$ to break one-bit RBAC scheme, then no algorithm has the advantage $\text{Adv}^{\text{RBAC–IND}}_{\Gamma,\mathcal{B}}(\lambda)$ to distinguish between $\text{Game}^{\mathcal{A}}_{v}$ and $\text{Game}^{\mathcal{A}}_{v+1}$, so that no algorithm has the advantage $k\ell \cdot \text{Adv}^{\text{RBAC–IND}}_{\Gamma,\mathcal{B}}(\lambda)$ to distinguish between $\text{Game}^{\mathcal{A}}_{\text{real3}}$ and $\text{Game}^{\mathcal{A}}_{k\ell}$. From the above deductions, we get $|\Pr[\text{Game}^{\mathcal{A}}_{\text{real}}(\lambda) = 1] - \Pr[\text{Game}^{\mathcal{A}}_{k\ell}(\lambda) = 1]| \le k\ell \cdot \text{Adv}^{\text{RBAC–IND}}_{\Gamma,\mathcal{B}}(\lambda) + k\ell \cdot \delta$. We also show that the simulator $\mathcal{S}$ runs identically to the $\text{Game}^{\mathcal{A}}_{k\ell}$, which means the $\text{Game}^{\mathcal{A}}_{k\ell}$ is distinguishable from the $\text{Game}^{\mathcal{S}}_{\text{sim}}$. Finally, we get $\text{Adv}^{\text{RBAC–UL}}_{\Gamma,\mathcal{S},\mathcal{A}}(\lambda) = |\Pr[\text{Game}^{\mathcal{A}}_{\text{real}}(\lambda) = 1] - \Pr[\text{Game}^{\mathcal{S}}_{\text{sim}}(\lambda) = 1]|$ which is defined in Definition 6.

6.5. Proof of Theorem 2. Let $\mathcal{A}$ be a RBAC-UL adversary against Γ^ℓ. We can construct a simulator $\mathcal{S}$ that runs Setup to generate PK and MSK. It runs $\mathcal{A}$ to answer the following queries. (1) When $\mathcal{A}$ outputs the set of EMRs on which it wishes to challenge, $\mathcal{S}$ then generates a set of EMR encapsulations where each encryption is an encryption of the all-zero message 0^ℓ and returns them to $\mathcal{A}$. (2) When $\mathcal{A}$ decides to corrupt some of these EMR encapsulations, the simulator $\mathcal{S}$ queries its own Corrupt procedure, learns the EMRs it needs to corrupt, and opens them bit-by-bit. If it needs to open an encapsulation component to a 0, it directly gives $\mathcal{A}$ the randomness it used when creating the encapsulation. Otherwise, $\mathcal{S}$ needs to open an encapsulation to a 1, it runs LeakToOne algorithm to find the randomness. (3) When $\mathcal{A}$ issues a secret credential query, $\mathcal{S}$ simply uses MSK to answer correctly. Through the above ways, the simulator $\mathcal{S}$ can generate the same output as $\mathcal{A}$. We use $\text{Game}_{\text{real}}$ and Game_{sim} to describe the games that $\mathcal{A}$ and $\mathcal{S}$ runs, respectively. Based on Definition 6, the target of Theorem 2 is to prove

$$\begin{aligned}&\left|\Pr\left[\text{Game}^{\mathcal{A}}_{\text{real}}(\lambda) = 1\right] - \Pr\left[\text{Game}^{\mathcal{S}}_{\text{sim}}(\lambda) = 1\right]\right| \\ &\quad \le k\ell \cdot \text{Adv}^{\text{RBAC–IND}}_{\Gamma,\mathcal{B}}(\lambda) + k\ell \cdot \delta.\end{aligned} \tag{34}$$

Lemma 7. *For any PPT algorithm $\mathcal{A}$, $Game^{\mathcal{A}}_{real}Adv_{\mathcal{A}}(\lambda) = Game^{\mathcal{A}}_{real1}Adv_{\mathcal{A}}(\lambda)$.*

Proof of Lemma 7. Since the randomness in $\text{Game}^{\mathcal{A}}_{\text{real}}$ and $\text{Game}^{\mathcal{A}}_{\text{real1}}$ are uniformly and independently chosen from Z_p, they are identically distributed from the view of $\mathcal{A}$. □

Lemma 8. *For any PPT algorithm $\mathcal{A}$, $Game^{\mathcal{A}}_{real1}Adv_{\mathcal{A}}(\lambda) = Game^{\mathcal{A}}_{real2}Adv_{\mathcal{A}}(\lambda)$.*

Proof of Lemma 8. If LeakToOne does not fail, its output is identically distributed as that in $\text{Game}^{\mathcal{A}}_{\text{real1}}$ from the view of $\mathcal{A}$'s. If LeakToOne does fail, $\text{Game}^{\mathcal{A}}_{\text{real2}}$ does the same operations as that in $\text{Game}^{\mathcal{A}}_{\text{real1}}$. □

Lemma 9. *Suppose that any execution of LeakToOne fails at most δ. For any PPT algorithm $\mathcal{A}$, the following holds:*

$$\left|\text{Game}^{\mathcal{A}}_{\text{real2}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}^{\mathcal{A}}_{\text{real3}}\text{Adv}_{\mathcal{A}}(\lambda)\right| \le k\ell \cdot \delta. \tag{35}$$

Proof of Lemma 9. Since there are at most $k\ell$ bits that have to be opened by LeakToOne algorithm in phase Corrupt, the worst event is that all the $k\ell$ bits are failed to be opened and

$\text{Game}_{\text{real3}}^{\mathcal{A}}$ does not make any response to the failure. The probability for the worst event is $k\ell \cdot \delta$. □

Lemma 10. *If a PPT adversary $\mathcal{B}$ can break the one-bit RBAC-IND scheme with $Adv_{\Gamma,\mathcal{B}}^{RBAC-IND}(\lambda)$, then there exists a PPT algorithm $\mathcal{A}$ so that*

$$\left|\text{Game}_{v}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{v+1}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda)\right| = Adv_{\Gamma,\mathcal{B}}^{\text{RBAC-IND}}(\lambda). \tag{36}$$

Proof of Lemma 10. Suppose $\text{Ev}_{v+1}^{1}(\text{Game}_{v}^{\mathcal{A}})$ is the event that in the execution of $\text{Game}_{v}^{\mathcal{A}}$, the $(v+1)$st bits sampled from the set of EMR are 1.

Let $\text{Game}_{v}^{\mathcal{A}}|\text{Ev}_{v+1}^{1}(\text{Game}_{v}^{\mathcal{A}})$ denote that $\text{Game}_{v}^{\mathcal{A}}$ is run in the condition that the event $\text{Ev}_{v+1}^{1}(\text{Game}_{v}^{\mathcal{A}})$ happens. Ev^{0} is the complementary event for Ev^{1}. Notice that in the event that the $(v+1)$st bit sampled in $\text{Game}_{v}^{\mathcal{A}}$ is 0, the game $\text{Game}_{v+1}^{\mathcal{A}}$ and $\text{Game}_{v}^{\mathcal{A}}$ are identical, because $\text{Game}_{v+1}^{\mathcal{A}}$ ignores the actual bit and encapsulates it as a bit 0 based on its definition, and $\text{Game}_{v}^{\mathcal{A}}$ encapsulates the actual $(v+1)$st bit which is 0. Thus, $\text{Ev}_{v+1}^{0}(\text{Game}_{v+1}) = \text{Ev}_{v+1}^{0}(\text{Game}_{v})$. Next, we compute $|\text{Game}_{v}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{v+1}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda)|$ as follows (we use G to represent Game):

$$\begin{aligned}
&\left|\text{Game}_{v}^{\mathcal{A}}\text{Adv}_{\mathcal{A}} - \text{Game}_{v+1}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}\right| \\
&= \left|\left(\Pr\left[G_{v}^{\mathcal{A}} \,\&\, \text{Ev}_{v+1}^{0}\left(G_{v}^{\mathcal{A}}\right)\right] + \Pr\left[G_{v}^{\mathcal{A}} \,\&\, \text{Ev}_{v+1}^{1}\left(G_{v}^{\mathcal{A}}\right)\right]\right)\right. \\
&\quad \left.-\left(\Pr\left[G_{v+1}^{\mathcal{A}} \,\&\, \text{Ev}_{v+1}^{0}\left(G_{v+1}^{\mathcal{A}}\right)\right] + \Pr\left[G_{v+1}^{\mathcal{A}} \,\&\, \text{Ev}_{v+1}^{1}\left(G_{v+1}^{\mathcal{A}}\right)\right]\right)\right| \\
&= \left|\left(\Pr\left[\text{Ev}_{v+1}^{0}\left(G_{v}^{\mathcal{A}}\right)\right] \cdot \Pr\left[G_{v}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{0}\left(G_{v}^{\mathcal{A}}\right)\right] - \left|\Pr\left[\text{Ev}_{v+1}^{0}\left(G_{v+1}^{\mathcal{A}}\right)\right] \cdot \Pr\left[G_{v}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{0}\left(G_{v+1}^{\mathcal{A}}\right)\right]\right)\right.\right. \\
&\quad \left.\left.+\left(\Pr\left[\text{Ev}_{v+1}^{1}\left(G_{v}^{\mathcal{A}}\right)\right] \cdot \Pr\left[G_{v}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{1}\left(G_{v}^{\mathcal{A}}\right)\right] - \Pr\left[\text{Ev}_{v+1}^{1}\left(G_{v+1}^{\mathcal{A}}\right)\right] \cdot \Pr\left[G_{v+1}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{1}\left(G_{v+1}^{\mathcal{A}}\right)\right]\right)\right|\right. \\
&= \left|\Pr\left[\text{Ev}_{v+1}^{0}\left(G_{v}^{\mathcal{A}}\right)\right] \cdot \left(\Pr\left[G_{v}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{0}\left(G_{v}^{\mathcal{A}}\right)\right] - \Pr\left[G_{v+1}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{0}\left(G_{v+1}^{\mathcal{A}}\right)\right]\right)\right. \\
&\quad \left.+\left(\Pr\left[\text{Ev}_{v+1}^{1}\left(G_{v}^{\mathcal{A}}\right)\right] \cdot \Pr\left[G_{v}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{1}\left(G_{v}^{\mathcal{A}}\right)\right] - \Pr\left[\text{Ev}_{v+1}^{1}\left(G_{v+1}^{\mathcal{A}}\right)\right] \cdot \Pr\left[G_{v+1}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{1}\left(G_{v+1}^{\mathcal{A}}\right)\right]\right)\right|.
\end{aligned} \tag{37}$$

Since in the event Ev_{v+1}^{0} both $\text{Game}_{v+1}^{\mathcal{A}}$ and $\text{Game}_{v}^{\mathcal{A}}$ are identical, the first item in the above formula is 0. It means that

$$\begin{aligned}
\left|\text{Game}_{v}^{\mathcal{A}}\text{Adv}_{\mathcal{A}} - \text{Game}_{v+1}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}\right| = \Big|&\Pr\left[\text{Ev}_{v+1}^{1}\left(\text{Game}_{v}^{\mathcal{A}}\right)\right] \cdot \Pr\left[\text{Game}_{v}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{1}\left(\text{Game}_{v}^{\mathcal{A}}\right)\right] \\
&-\Pr\left[\text{Ev}_{v+1}^{1}\left(\text{Game}_{v+1}^{\mathcal{A}}\right)\right] \cdot \Pr\left[\text{Game}_{v+1}^{\mathcal{A}} \,\middle|\, \text{Ev}_{v+1}^{1}\left(\text{Game}_{v+1}^{\mathcal{A}}\right)\right]\Big|.
\end{aligned} \tag{38}$$

Next, we consider the adversary $\mathcal{B}$ against the one-bit RBAC scheme with 1SPL. $\mathcal{B}$ runs $\mathcal{A}$ while simulating its RBAC-IND environment as in either $\text{Game}_{v}^{\mathcal{A}}|\text{Ev}_{v+1}^{1}(\text{Game}_{v}^{\mathcal{A}})$ or $\text{Game}_{v+1}^{\mathcal{A}}|\text{Ev}_{v+1}^{1}(\text{Game}_{v+1}^{\mathcal{A}})$. Note that $\text{Challenge}_{\text{B}}$ is used to denote that the adversary $\mathcal{B}$ runs the Challenge phase in the RBAC-IND experiment.

Setup: $\mathcal{B}$ generates a public key PK by running the system setup algorithm, and it sends PK to $\mathcal{A}$.

Query Phase 1: $\mathcal{A}$ issues a secret credential query for the medical staff associated with role $\overrightarrow{R}$. $\mathcal{B}$ creates the secret credential by running the credential generation algorithm and return the secret credential to $\mathcal{A}$.

Challenge: the adversary $\mathcal{A}$ outputs a set of EMR files $\overrightarrow{\text{EMR}} = \{M_1, M_2, \ldots, M_k\}$ and a challenge access policy set $P^* = \{P_1^*, P_2^*, \ldots, P_k^*\}$ to $\mathcal{B}$. Each challenge access policy P_i^* in the set should satisfy that for all the secret credential queries for $\overrightarrow{R}$ issued in Query Phase 1, $\overrightarrow{R} \notin \text{Pref}(P_i^*)$. We note that each EMR file EMR_i for $i \in [1,k]$ constitutes of ℓ bits since we let the Γ^{ℓ} be a ℓ-bit RBAC scheme. $\mathcal{B}$ randomly chooses $i \xleftarrow{R} [1,\ell]$ and $j \xleftarrow{R} [1,k]$. For each $i \in [1,\ell]$ and $j \in [1,k]$, we consider three cases:

(i) When $(i-1)\cdot\ell + j \le v$, $\text{EMR}[i][j]$ is ignored and replaced by 0. $\mathcal{B}$ randomly chooses $r[i][j] \xleftarrow{R} Z_p$, computes

$\mathrm{En}[i][j] \xleftarrow{R} \mathrm{EMREnc}(\mathrm{PK}, P_i^*, 0, r[i][j])$, and returns it to $\mathscr{A}$.

(ii) When $(i-1)\cdot\ell + j = v+1$, $\mathscr{B}$ encrypts the $v+1$st bit in two different conditions. If $\mathrm{EMR}[i][j] = 1$, $\mathscr{B}$ runs the $\mathrm{Challenge}_{\mathscr{B}}$ algorithm against the RBAC-IND scheme. Specifically speaking, $\mathscr{B}$ flips a random coin to decide whether it encapsulates 0 or 1 under the challenge access policy P_i^*. Then $\mathscr{B}$ returns the cihpertext to $\mathscr{A}$. If $\mathrm{EMR}[i][j] = 0$, $\mathrm{EMR}[i][j]$ is encapsulated normally. $\mathscr{B}$ randomly chooses $r[i][j] \xleftarrow{R} Z_p$, executes $\mathrm{En}[i][j] \xleftarrow{R} \mathrm{EMREnc}(\mathrm{PK}, P_i^*, \mathrm{EMR}[i][j], r[i][j])$ and returns it to $\mathscr{A}$.

(iii) When $(i-1)\cdot\ell + j > v+1$, $\mathrm{EMR}[i][j]$ is encapsulated normally. $\mathscr{B}$ randomly$_R$ chooses $r[i][j] \xleftarrow{R} Z_p$, computes $\mathrm{En}[i][j] \xleftarrow{} \mathrm{EMREnc}(PK, P_i^*, \mathrm{EMR}[i][j], r[i][j])$, and returns it to $\mathscr{A}$.

Corrupt: $\mathscr{A}$ outputs a set $I \subseteq [1,n]$ and then $\mathscr{B}$ learns $\overrightarrow{\mathrm{EMR}}[I]$. For each index $i \in I$ and each $j \in [1,\ell]$, $\mathscr{B}$ generates the randomness as follows:

(i) If $\mathrm{EMR}[i][j] = 0$, $\mathscr{B}$ returns the actual randomness it used to generated En_i.

(ii) If $\mathrm{EMR}[i][j] = 1$, $\mathscr{B}$ runs LeakToOne algorithm to get randomness under which the EMREnc algorithm applied to 1 would produce $\mathrm{En}_{[i][j]}$, namely,

$$r[i][j] \xleftarrow{R} \text{Leak to one}\left(\mathrm{PK}, P_i^*, \mathrm{En}_{[i][j]}\right). \tag{39}$$

Finally, $\mathscr{B}$ returns $(M[i][j], r[i][j])_{i\in[1,k]\,j\in[1,\ell]}$ to $\mathscr{A}$.

Query Phase 2: Query Phase 1 is repeated adaptively except that $\overrightarrow{R} \notin \mathrm{Pref}(P_i^*)$.

Output: when $\mathscr{A}$ halts with out, $\mathscr{B}$ halts and outputs $(\overrightarrow{\mathrm{EMR}}, P^*, I, \mathrm{out})$.

The adversary $\mathscr{B}$ only runs $\mathrm{Challenge}_{\mathscr{B}}$ in the event Ev_{v+1}^1 $(\mathrm{EMR}[i][j] = 1)$ in the Challenge phase, such that all of its advantage comes from this case. It means that $\Pr[\mathrm{Ev}_{v+1}^1(\mathrm{Game}_v)] = 1$ and $\Pr[\mathrm{Ev}_{v+1}^1(\mathrm{Game}_{v+1})] = 1$. It is important to notice that in the event Ev_{v+1}^1, if $\mathscr{B}$ decides to encapsulate 1, it simulates its environment as in playing $\mathrm{Game}_v^{\mathscr{A}}$ with $\mathscr{A}$. If $\mathscr{B}$ decides to encapsulate 0, it simulates the environment as in $\mathrm{Game}_{v+1}^{\mathscr{A}}$. Therefore, the advantage of $\mathscr{A}$ to distinguish $\mathrm{Game}_v^{\mathscr{A}} | \mathrm{Ev}_{v+1}^1(\mathrm{Game}_v^{\mathscr{A}})$ and $\mathrm{Game}_{v+1}^{\mathscr{A}} | \mathrm{Ev}_{v+1}^1(\mathrm{Game}_{v+1}^{\mathscr{A}})$ depends on the advantage of $\mathscr{B}$ to distinguish that the challenge EMR-encapsulation is for 1 or 0. It is easy to see that

$$\begin{aligned}\mathrm{Adv}_{\Gamma,\mathscr{B}}^{\mathrm{RBAC-IND}}(\lambda) = &\left|\Pr\left[\mathrm{Game}_v^{\mathscr{A}} \,\middle|\, \mathrm{Ev}_{v+1}^1\left(\mathrm{Game}_v^{\mathscr{A}}\right)\right]\right. \\ &\left. - \Pr\left[\mathrm{Game}_{v+1}^{\mathscr{A}} \,\middle|\, \mathrm{Ev}_{v+1}^1\left(\mathrm{Game}_{v+1}^{\mathscr{A}}\right)\right]\right|.\end{aligned} \tag{40}$$

Combined with equation (38), we get

$$\left|\mathrm{Game}_v^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda) - \mathrm{Game}_{v+1}^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda)\right| = \mathrm{Adv}_{\Gamma,\mathscr{B}}^{\mathrm{RBAC-IND}}(\lambda). \tag{41}$$

Furthermore, since

$$\begin{aligned}&\left|\mathrm{Game}_0^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda) - \mathrm{Game}_{k\ell}^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda)\right| \\ &= \sum_{v=0}^{k\ell-1}\left(\left|\mathrm{Game}_v^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda) - \mathrm{Game}_{v+1}^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda)\right|\right),\end{aligned} \tag{42}$$

we conclude

$$\left|\mathrm{Game}_0^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda) - \mathrm{Game}_{k\ell}^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda)\right| = k\ell \cdot \mathrm{Adv}_{\Gamma,\mathscr{B}}^{\mathrm{RBAC-IND}}(\lambda). \tag{43}$$

□

Lemma 11. *For any PPT algorithm $\mathscr{A}$,*

$$\mathrm{Game}_{k\ell}^{\mathscr{A}}\mathrm{Adv}_{\mathscr{A}}(\lambda) = \mathrm{Game}_{\mathrm{sim}}^{\mathscr{S}}\mathrm{Adv}_{\mathscr{S}}(\lambda). \tag{44}$$

Proof of Lemma 11. First, we compare two games $\mathrm{Game}_{k\ell}^{\mathscr{A}}$ and $\mathrm{Game}_{\mathrm{sim}}^{\mathscr{S}}$.

$\mathrm{Game}_{k\ell}^{\mathscr{A}}$ works as follows:

Setup: $\mathscr{A}$ receives a public key from the challenger $\mathscr{C}$.

Query Phase 1: $\mathscr{A}$ issues a secret credential query for the medical staff associated with role $\overrightarrow{R}$. The challenger $\mathscr{C}$ creates the secret credential by running the credential generation algorithm and returns the secret credential to $\mathscr{A}$.

Challenge: the adversary $\mathscr{A}$ outputs a set of EMR files $\overrightarrow{\mathrm{EMR}} = \{\mathrm{EMR}_1\mathrm{EMR}_2, \ldots, \mathrm{EMR}_k\}$ and the challenge access policy set $P^* = \{P_1^*, P_2^*, \ldots, P_k^*\}$ to the challenger $\mathscr{C}$. Each challenge access policy P_i^* in the set should satisfy that for all the access credential queries for $\overrightarrow{R}$ issued in Query Phase 1, $\overrightarrow{R} \notin \mathrm{Pref}(P_i^*)$. $\mathscr{C}$ randomly chooses elements $r_1, r_2, \ldots, r_k \xleftarrow{R} Z_p$ where $p = |\mathbb{G}|$. We denote $\overrightarrow{r} = \{r_1, r_2, \ldots, r_k\}$. $\mathscr{C}$ ignores the input EMRs and regards the components as all-0 messages. Then it encrypts each 0^ℓ message as follows:

$$\mathrm{En}_i \xleftarrow{R} \mathrm{Enc}\left(\mathrm{PK}, P_i^*, 0^\ell, r_i\right), \tag{45}$$

$\mathscr{C}$ returns the set of EMR encapsulation $\overrightarrow{\mathrm{En}} = \{\mathrm{En}_1, \mathrm{En}_2, \ldots, \mathrm{En}_k\}$ to $\mathscr{A}$.

Corrupt: $\mathscr{A}$ outputs a set $I \subseteq [1,n]$ and then $\mathscr{C}$ opens the corresponding ciphertext to get $\overrightarrow{\mathrm{EMR}}[I]$. For each index $i \in I$ and each $j \in [1,\ell]$, $\mathscr{C}$ makes the randomness as follows:

(i) If $\mathrm{EMR}[i][j] = 0$, $\mathscr{C}$ returns the actual randomness it used to generated En_i.

(ii) If $\mathrm{EMR}[i][j] = 1$, $\mathscr{C}$ runs LeakToOne to get the randomness used by EMREnc to compute $\mathrm{En}_{[i][j]}$ when encrypting 1, namely, $r[i][j] \xleftarrow{R}$ Leak to one $(\mathrm{PK}, P_i^*, \mathrm{En}_{[i][j]})$.

Finally, $\mathscr{C}$ returns $(\mathrm{EMR}[i][j], r[i][j])_{i\in[,k],j\in[1,\ell]}$ to $\mathscr{A}$.

Query Phase 2: Query Phase 1 is repeated adaptively except that $\overrightarrow{R} \notin \mathrm{Pref}(P_i^*)$.

Output: When the adversary $\mathscr{A}$ halts with out, $\mathscr{C}$ halts and outputs $(\overrightarrow{\mathrm{EMR}}, P^*, I, \mathrm{out})$.

$\text{Game}_{\text{sim}}^{\mathcal{S}}$ works as follows:

Setup: $\mathcal{S}$ generates the public key PK by running the system setup algorithm and then sends the public key to $\mathcal{A}$.

Query Phase 1: $\mathcal{A}$ requests the secret credential for the medical staff associated with role $\overrightarrow{R}$. $\mathcal{S}$ creates the secret credential by running the credential generation algorithm and return the secret credential to $\mathcal{A}$.

Challenge: $\mathcal{A}$ outputs a set of EMR $\overrightarrow{\text{EMR}} = \{M_1, M_2, \ldots, M_k\}$ and the challenge access policy set $P^* = \{P_1^*, P_2^*, \ldots, P_k^*\}$ to $\mathcal{B}$. Each challenge access policy P_i^* in the set should satisfy that for all the access credential queries for $\overrightarrow{R}$ issued in Query Phase 1, $\overrightarrow{R} \notin \text{Pref}(P_i^*)$. Note that each EMR_i for $i \in [1, k]$ consists of ℓ bits since we let Γ^ℓ be a ℓ-bit RBAC scheme. $\mathcal{S}$ randomly chooses elements $r_1, r_2, \ldots, r_k \xleftarrow{R} Z_p$ where $p = |\mathbb{G}|$. We denote $\overrightarrow{r} = \{r_1, r_2, \ldots, r_k\}$. $\mathcal{S}$ ignores the input EMRs and regards the components as all-0 messages. Then it encapsulates each 0^ℓ message as $\text{En}_i \xleftarrow{R} \text{Enc}(\text{PK}, P_i^*, 0^\ell, r_i)$. Finally, $\mathcal{S}$ returns the EMR encapsulations $\overrightarrow{\text{En}} = \{\text{En}_1, \text{En}_2, \ldots, \text{En}_k\}$ to $\mathcal{A}$.

Corrupt: $\mathcal{A}$ outputs a set $I \subseteq [1, n]$ and then $\mathcal{S}$ learns $\overrightarrow{\text{EMR}}[I]$. For each index $i \in I$ and each $j \in [1, \ell]$, $\mathcal{S}$ makes the randomness as follows:

(i) If $\text{EMR}[i][j] = 0$, $\mathcal{S}$ returns the actual randomness used to generate En_i.

(ii) If $\text{EMR}[i][j] = 1$, $\mathcal{S}$ runs LeakToOne to get randomness used by EMREnc to compute $\text{En}_{[i][j]}$ for encapsulation of 1, namely, $r[i][j] \xleftarrow{R} \text{Leak to one}(\text{PK}, P_i^*, 0^\ell, \text{En}_{[i][j]})$.

Finally, $\mathcal{S}$ returns $(\text{EMR}[i][j], r[i][j])_{i \in [1,k], j \in [1,\ell]}$ to $\mathcal{A}$.

Query Phase 2: Query Phase 1 is repeated adaptively except that $\overrightarrow{R} \notin \text{Pref}(P_i^*)$.

Output: when $\mathcal{A}$ halts with output out, $\mathcal{S}$ halts and outputs $(\overrightarrow{\text{EMR}}, P^*, I, \text{out})$.

$\mathcal{S}$ queries its own Corrupt procedure on I and learns $\overrightarrow{\text{EMR}}[I]$ instead of getting them directly as in $\text{Game}_{k\ell}^{\mathcal{A}}$. From the view of the adversary $\mathcal{A}$, there is no difference of the corrupted EMRs and the sampled randomness. Therefore, $\mathcal{S}$ runs identically with $\text{Game}_{k\ell}^{\mathcal{A}}$. □

Table 4: The efficiency of the proposed scheme.

	The atom roles is n
MSK size	$n + 3$
$\text{SC}^{\overrightarrow{R}}$ size	$n + 2 - \|\|\overrightarrow{R}\|\|$
SCGen time	$(2n - \|\|\overrightarrow{R}\| + 4)t_e + (n + 2)t_m$
SCDeleg time	$(2n - \|\|\overrightarrow{R}\|\| + 5)t_e + (n + 5)t_m$
EMREnc time	$(\|P\| + 3)t_e + (\|P\| + 2)t_m$
EMRDec time	$2t_p$

Proof of Theorem 2. From the above analysis, the simulator $\mathcal{S}$ described in $\text{Game}_{\text{sim}}^{\mathcal{S}}$ runs identically to $\text{Game}_{k\ell}^{\mathcal{A}}$. So we have $\text{Game}_{k\ell}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) = \text{Game}_{\text{sim}}^{\mathcal{S}}\text{Adv}_{\mathcal{S}}(\lambda)$. Combining all the above lemmas, we get

$$\begin{aligned}
&\left|\text{Game}_{\text{real}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{sim}}^{\mathcal{S}}\text{Adv}_{\mathcal{S}}(\lambda)\right| \\
&\le \left|\text{Game}_{\text{real}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{real1}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda)\right| + \left|\text{Game}_{\text{real1}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{real2}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda)\right| \\
&\quad + \left|\text{Game}_{\text{real2}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{real3}}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda)\right| + \sum_{v=0}^{k\ell-1}\left(\left|\text{Game}_{v}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{v+1}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda)\right|\right) \\
&\quad + \left|\text{Game}_{k\ell}^{\mathcal{A}}\text{Adv}_{\mathcal{A}}(\lambda) - \text{Game}_{\text{sim}}^{\mathcal{S}}\text{Adv}_{\mathcal{S}}(\lambda)\right| \le k\ell \cdot \delta + k\ell \cdot \text{Adv}_{\Gamma,\mathcal{B}}^{\text{RBAC-IND}}(\lambda).
\end{aligned} \tag{46}$$

According to Definition 6, we get $\text{Adv}_{\Gamma^\ell,\mathcal{S},\mathcal{A}}^{\text{RBAC-UL}}(\lambda) \le k\ell \cdot \text{Adv}_{\Gamma,\mathcal{B}}^{\text{RBAC-IND}}(\lambda) + k\ell \cdot \delta$, which proves Theorem 2. □

7. Performance Analyses

7.1. Improve User Experience. To achieve better user experience, we speed up credential generation and EMR encapsulation by applying online/offine cryptography [38]. The offline phase executes most of heavy computations by assuming a set of random roles, while the online phase only performs light computations to produce the EMR encapsulation and the secret credential once the true roles are available. "Ours&RBAC" is denoted as the scheme with improved efficiency.

7.2. Theoretical Analysis. Table 4 shows the efficiency of the proposed one-bit RBAC scheme. We denote t_e as one exponent operation time, t_m as one multiplication time, and t_p as one pairing operation time. The maximal depth of the hierarchy for a access policy is $\|P\|$. $\|\overrightarrow{R}\|$ is the number of atom roles in a secret credential. In the procedure of SCGen, SCDeleg, EMREnc, and EMRDec, exponentiations can be precomputed by choosing the random exponents.

Table 5 compares several schemes in different perspectives. The properties of scalable sharing, flexible access, and leakage controllability support further rendering our scheme with improved efficiency to practice.

Table 5: Comparison with related work.

	SCGen time	EMREnc time	Parings in EMRDec	Leakage controllability	Scalable sharing	Flexibility
[12]	$(1+4\|\vec{R}\|)t_e\,(1+4\|\|\vec{R}\|\|)t_m$	$6t_e + t_m + t_p + t_p$	$3\|\vec{R}\|$	×	√	√
[16]	$3(n+4)t_e + (3\|\|\vec{R}\|\|+4)t_m$	$(\|P\|+4)t_e + (\|\|P\|\|+4)t_m$	3	×	√	√
[29]	$5t_e + 3t_m$	$5t_e + 3t_m$	2	√	×	×
[39]	$2(\|\|\vec{R}\|\|+1)t_e + \|\vec{R}\|t_m$	$5t_e + \|P\|t_m + t_p$	2	×	√	√
[40]	$2(\|\vec{R}\|+1)t_e + \|\vec{R}\|t_m$	$4t_e + t_m + t_p$	$2\|\vec{R}\|$	×	×	√
Ours	$(2n-\|\vec{R}\|+4)t_e + (n+2)t_m$	$(\|\|P\|\|+3)t_e + (\|P\|+2)t_m$	2	√	√	√
Ours & RBAC	$\|\vec{R}\| \cdot t_m$	$\|P\| \cdot t_m$	2	√	√	√

Table 6: A single computation execution time.

	t_e	t_m	t_p
Prime order bilinear group	4.62 ms	0.04 ms	38.56 ms
Composite order bilinear group	130 ms	0.16 ms	148.52 ms

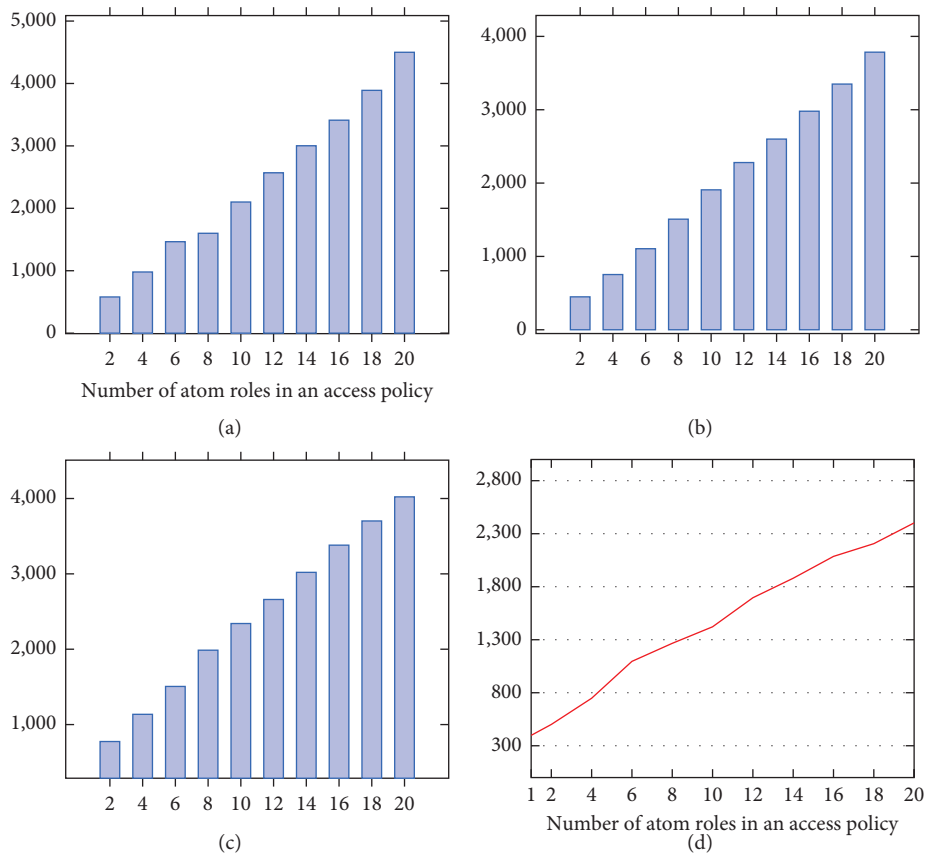


Figure 2: Continued.

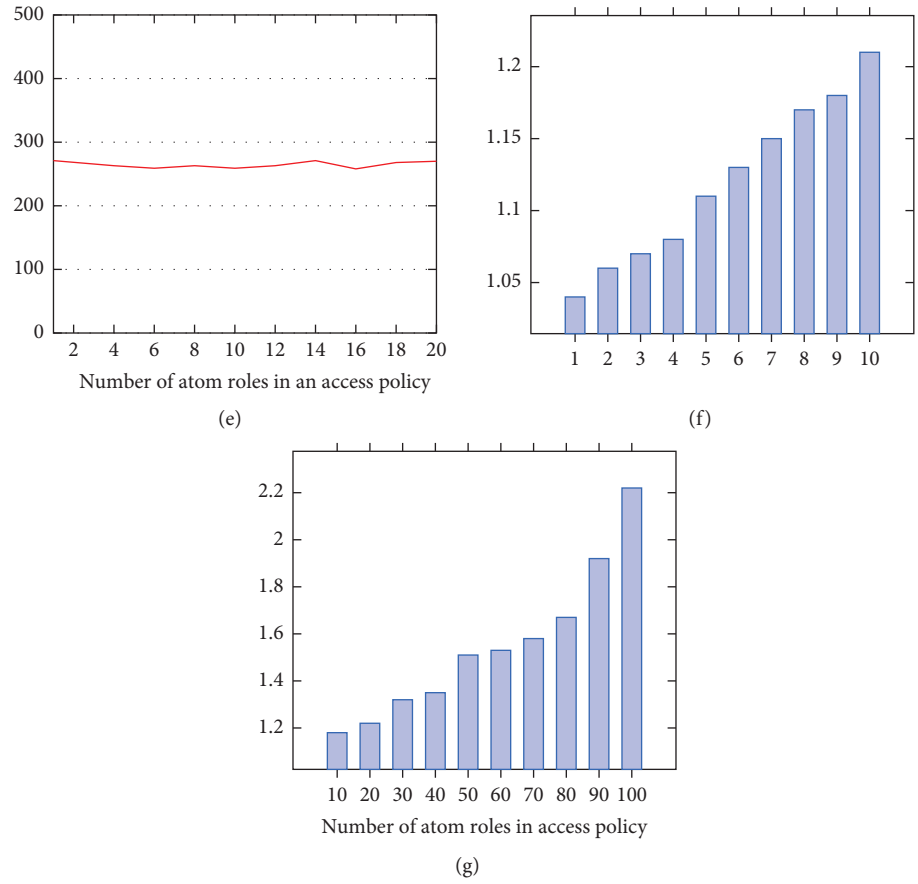


Figure 2: Experimental results for our proposed system. (a) System setup time (ms). (b) SC generation time (ms). (c) SC delegation time (ms). (d) Encapsulation time (ms). (e) Decapsulation time (ms). (f) Improved SC generation time (ms). (g) Improved encapsulation time (ms).

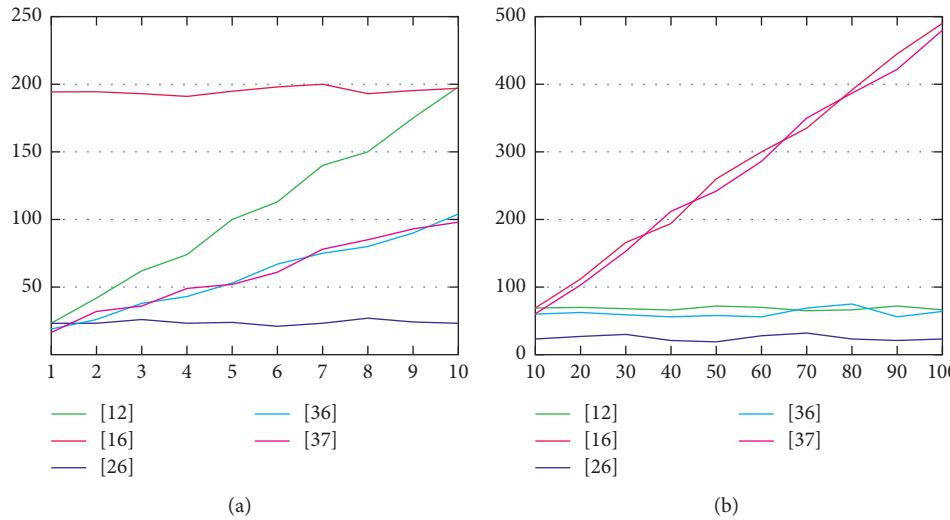


Figure 3: Experimental result for the compared related work. (a) SC Generation time for the related work (ms). (b) EHR Encapsulation time for the related work (ms).

7.3. Performance Analysis. We conduct experiment on Intel Core i7 processor with 8 GB RAM and 2.6 GHZ CPU. We use elliptic curve type A1 for the Tate symmetric pairing. Both group order of $\mathbb{Z}_N$ and element size in $\mathbb{G}$ are configured as 512 bits. The experiment is executed with jPBC library. We test the single computation execution times t_e, t_m, and t_p for the prime order bilinear group and the composite order bilinear group, which are used in the

related work and our work separately. Table 6 shows the compared running time.

We also test the operational time for system setup, credential generation, delegation, EMR encapsulation, and decapsulation for our system, as Figure 2 illustrates. Figure 2(f) and 2(g) show the operational time when user experience is improved.

Figures 3(a) and 3(b) show the operational time for the compared related work, where prime order bilinear groups are used. The computation of SC generation time and EHR encapsulation time shows superior efficiency when compared with our work without performance improved. That is why we apply the performance improvement algorithm in our system, so as to ensure both efficiency and security. The Y-axis represents the operational time in milliseconds. The X-axis in Figures 2(b), 2(c), 2(f), and 3(a) means the number of related atom roles included in a role of medical staff. The X-axis in Figures 2(a), 2(d), 2(e), 2(g), and 3(b) means the number of atom roles in an access policy.

8. Conclusion

We consider a multiparty communication scenario in a medical cloud storage system. A lot of medical records are outsourced on the cloud and accessed by medical staff with hierarchical privileges. We summarize different adversarial behaviours and construct a RBAC-UL scheme against many kinds of leakages. Performance analyses show that our scheme has advantages in scalability, flexibility, and the controllability of privacy leakage.

Data Availability

No specific data are available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFB1400702), the National Natural Science Foundation of China (61972017, 61972018, 61932014), the Beijing Natural Science Foundation (4182033), the National Cryptography Development Fund (MMJJ20180215), and the Special Scientific Research for Civil Aircraft of Ministry of Industry and Information Technology.

References

[1] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM Conference on Computer and Communications Security CCS 2008*, pp. 417–426, ACM, Alexandria, VA, USA, October 2008.

[2] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[3] M. Bellare, A. Desai, and D. Pointcheval, "Relations among notions of security for public-key encryption schemes," in *CRYPTO'98*, vol. 1462, pp. 26–45, Springer, Berlin, Germany, 1998.

[4] V. T. Hoang, J. Katz, A. O'Neill, and M. Zaheri, "Selective-opening security in the presence of randomness failures," in *Advances in Cryptology—ASIACRYPT 2016*, vol. 10032, pp. 278–306, Springer, Berlin, Germany, 2016.

[5] B. Mihire and B. Tackmann, "Nonce-based cryptography: retaining security when randomness fails," in *EUROCRYPT*, vol. 9665, pp. 729–757, Springer, Berlin, Germany, 2016.

[6] M. R. Albrecht and K. G. Paterson, "Lucky microseconds: a timing attack on Amazon's s2n implementation of TLS," in *Advances in Cryptology—EUROCRYPT 2016*, vol. 9665, pp. 622–643, Springer, Berlin, Germany, 2016.

[7] M. C. Mont, P. Bramhall, and K. Harrison, "A flexible role-based secure messaging service: exploiting IBE technology for privacy in health care," in *Proceedings of the International Workshop on Database and Expert Systems Applications*, pp. 432–437, IEEE Computer Society, Prague, Czech Republic, September 2003.

[8] Z. Zhang, H. Wang, and A. V. Vasilakos, "ECG-cryptography and authentication in body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1070–1078, 2012.

[9] Z. Qin, K. Feng, S. Hu et al., "A novel identity-based security scheme for wireless sensor networks," in *Proceedings of the Tenth International Conference on Computational Intelligence and Security*, pp. 662–666, IEEE, Kunming, China, November 2014.

[10] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," in *Proceedings of the ACM Conference on Wireless Network Security*, pp. 148–153, ACM, Alexandria, VA, USA, March 2008.

[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 2006*, pp. 89–98, ACM, Incheon, South Korea, October 2006.

[12] Z. Wan, J. E. Liu, and R. H. Deng, "HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[13] Y. L. Tan, B. M. Goi, R. Komiya, and S. Y. Tan, "A study of attribute-based encryption for body sensor networks," in *International Conference on Informatics Engineering and Information Science*, vol. 251, pp. 238–247, Springer, Berlin, Germany, 2011.

[14] R. Gandikota, R. Eswara, and J. Appawala, "Fine-grained access control of EHRs in cloud using CP-ABE with user revocation," *Health and Technology*, vol. 9, no. 4, pp. 487–496, 2019.

[15] C. Guo, R. Zhuang, Y. Jie, Y. Ren, T. Wu, and K.-K. R. Choo, "Fine-grained database field search using attribute-based encryption for e-healthcare clouds," *Journal of Medical Systems*, vol. 40, no. 11, p. 235, 2016.

[16] W. Liu, X. Liu, J. Liu, Q. Wu, J. Zhang, and Y. Li, "Auditing and revocation enabled role-based access control over outsourced private EHRs," in *Proceedings of the 2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC 2015)*, pp. 336–341, IEEE, New York, NY, USA, August 2015.

[17] X. Zhou, J. Liu, W. Liu, and Q. Wu, "Anonymous role-based access control on e-health records," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIACCS 2016)*, pp. 559–570, ACM, Xi'an, China, May 2016.

[18] W. Liu, J. Liu, Q. Wu, and B. Qin, "Hierarchical identity-based broadcast encryption," in *Information Security and Privacy*, vol. 8544, pp. 242–257, Springer, Cham, Switzerland, 2014.

[19] L. Qin, V. Varadharajan, and K. Gopinath, "A secure role-based cloud storage system for encrypted patient-centric health records," *The Computer Journal*, vol. 59, no. 11, pp. 1593–1611, 2016.

[20] J. Liu, W. Liu, Q. Wu, and X. Liu, "Auditing revocable privacy-preserving access control for EHRs in clouds," *The Computer Journal*, vol. 60, no. 12, pp. 1125–1132, 2017.

[21] G. Ramu and A. Jayanthi, "Enhancing medical data security in the cloud using RBAC-CPABE and ASS," *International Journal of Applied Engineering Research*, vol. 13, no. 7, pp. 21–25, 2018.

[22] B. Qin and S. Liu, "Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter," in *Advances in Cryptology—ASIACRYPT 2013*, vol. 8270, pp. 381–400, Springer, Berlin, Germany, 2013.

[23] B. Qin and S. Liu, "Leakage-flexible CCA-secure public-key encryption: simple construction and free of pairing," in *Public-Key Cryptography—PKC 2014*, vol. 8383, pp. 19–36, Springer, Berlin, Germany, 2014.

[24] S. Yilek, "Resettable public-key encryption: how to encrypt on a virtual machine," in *Topics in Cryptology—CT-RSA 2010*, vol. 5985, pp. 41–56, Springer, Berlin, Germany, 2010.

[25] K. G. Paterson, J. C. N. Schuldt, and D. L. Sibborn, "Related randomness attacks for public key encryption," in *Public-Key Cryptography—PKC 2014*, vol. 8383, pp. 465–482, Springer, Berlin, Germany, 2014.

[26] F. Serge, H. Dennis, K. Eike, and W. Hoeteck, "Encryption schemes secure against chosen-ciphertext selective opening attacks," in *EUROCRYPT*, vol. 6110, pp. 381–402, Springer, Berlin, Germany, 2010.

[27] B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud, "Lossy encryption: constructions from general assumptions and efficient selective opening chosen ciphertext security," in *Lecture Notes in Computer Science*, vol. 7073, pp. 70–88, Springer, Berlin, Germany, 2011.

[28] Z. Zhang, S. S. M. Chow, and Z. Cao, "Post-challenge leakage in public-key encryption," *Theoretical Computer Science*, vol. 572, pp. 25–49, 2015.

[29] M. Bellare, B. Waters, and S. Yilek, "Identity-based encryption secure against selective opening attack," in *TCC 2011*, vol. 6597, pp. 321–334, Springer, Berlin, Germany, 2011.

[30] J. Lai, R. H. Deng, S. Liu, J. Weng, and Y. Zhao, "Identity-based encryption secure against selective opening chosen-ciphertext attack," in *Advances in Cryptology—EUROCRYPT 2014*, vol. 8441, pp. 77–92, Springer, Berlin, Germany, 2014.

[31] Y. Chen, Z. Zhang, D. Lin, and Z. Cao, "Generalized (identity-based) hash proof system and its applications," *Security and Communication Networks*, vol. 9, no. 12, pp. 1698–1716, 2016.

[32] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography 2005*, vol. 3378, pp. 325–341, Springer, Berlin, Germany, 2005.

[33] C. Gentry, "Practical identity-based encryption without random oracles," in *EUROCRYPT 2006*, pp. 445–464, Springer, Berlin, Germany, 2006.

[34] C. Gentry and S. Halevi, "Hierarchical identity based encryption with polynomially many levels," in *Theory of Cryptography 2009*, vol. 5444, pp. 437–456, Springer, Berlin, Germany, 2009.

[35] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Advances in Cryptology—EUROCRYPT 2002*, vol. 2332, pp. 466–481, Springer, Berlin, Germany, 2002.

[36] R. Canetti, C. Dwork, M., and R. Ostrovsky, "Deniable encryption," *Advances in Cryptology—CRYPTO'97*, vol. 1294, pp. 90–104, Springer, Berlin, Germany, 1997.

[37] A. Naor and B. Waters, "New techniques for dual system encryption and fully secure hibe with short ciphertexts," in *Theory of Cryptography 2010*, vol. 5978, pp. 455–479, Springer, Berlin, Germany, 2010.

[38] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography—PKC 2014*, vol. 8383, pp. 293–310, Springer, Berlin, Germany, 2014.

[39] L.-Y. Yeh, P.-Y. Chiang, Y.-L. Tsai, and J.-L. Huang, "Cloud-based fine-grained health information access control framework for lightweight IoT devices with dynamic auditing and attribute revocation," *IEEE Transactions on Cloud Computing*, vol. 6, no. 2, pp. 532–544, 2018.

[40] M. Barua, X. Liang, R. Lu, and X. Shen, "ESPAC: enabling security and patient-centric access control for ehealth in cloud computing," *International Journal of Security and Networks*, vol. 6, no. 2-3, pp. 67–76, 2011.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 1315047, 9 pages
https://doi.org/10.1155/2019/1315047

Research Article

An API Semantics-Aware Malware Detection Method Based on Deep Learning

Xin Ma, Shize Guo, Wei Bai, Jun Chen, Shiming Xia, and Zhisong Pan

Command and Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China

Correspondence should be addressed to Zhisong Pan; hotpzs@hotmail.com

Received 19 February 2019; Revised 28 April 2019; Accepted 20 May 2019; Published 11 November 2019

Guest Editor: Wenjia Li

The explosive growth of malware variants poses a continuously and deeply evolving challenge to information security. Traditional malware detection methods require a lot of manpower. However, machine learning has played an important role on malware classification and detection, and it is easily spoofed by malware disguising to be benign software by employing self-protection techniques, which leads to poor performance for existing techniques based on the machine learning method. In this paper, we analyze the local maliciousness about malware and implement an anti-interference detection framework based on API fragments, which uses the LSTM model to classify API fragments and employs ensemble learning to determine the final result of the entire API sequence. We present our experimental results on Ali-Tianchi contest API databases. By comparing with the experiments of some common methods, it is proved that our method based on local maliciousness has better performance, which is a higher accuracy rate of 0.9734.

1. Introduction

Malware remains a serious problem for individuals, corporations, and nation information security all the time, as attackers continuously use it as a tool to get illegal profit and damage information infrastructure. Just in the first half of 2018 alone, the 360 Internet Security Center has intercepted 140 million new malicious programs, and an average of 795,000 new malicious programs was intercepted every day. We are seriously facing malicious code attacks all the time. Traditional signature-based feature detection methods, which take a lot of manpower and require professional knowledge, are difficult to combat. In fact, a lot of malware come from the benign software which was infected by malicious code snippets. Malware authors even use polymorphism to reorder these codes and create several malicious variations. By modification, this malware is hardly recognized by antivirus software. This is the key point that traditional detection technology for malware detection should be explored.

With the successful application of deep learning in the fields of image processing, speech recognition, and software engineering, machine learning has become an important method for analyzing malware in the last 10 years. As the Kaspersky Lab said, deep learning is a special machine learning approach that facilitates the features extraction method to get a high level of abstraction from low-level data. As the machine learning outperforms a great power for data handling, many scholars propose a series of methods using machine learning to detect malware. Previous studies [1, 2] used image visualization to classify malware by the similarity of texture. The studies [3, 4] are based on API call sequence to detect malware. In [5], the researcher uses LSTM and CNN to build a learning model for malware classification and detection, which transforms ASM code to sequence and uses NLP method to handle it. Above all, although these methods have made a stride for malware classification and detection, there are still many flaws, leading to attack from the malware writer, such as self-extraction and obfuscation. The key point is that these methods of extracting features are easily disturbed. Therefore, facing a lot of confrontation, it performs badly.

In this paper, we propose an effective detection framework based on API fragments, which can be employed

on the Windows platform by extracting the API execution sequence from executable files. Firstly, we analyze the local malicious characteristics of malicious code. On this basis, we cut the API execution sequence into API fragments and train the model using API fragments. Finally, we use ensemble learning to ultimately make a decision based on the performance of each segment of the entire sequence. The results show that this method can effectively extract key features and give a good classification result. Our work contributes as follows:

(1) Analyze the important meaning of local maliciousness in the API execution sequence

(2) Implement an API fragment-based detection framework with strong anti-interference using LSTM model and ensemble learning

(3) Expound the meaning of the parameters and obtain an optimal value

2. Related Work

The field of malicious code classification and detection is currently divided into traditional methods and machine learning methods. The traditional methods rely on a large amount of expert knowledge to extract the malicious features by reverse analyzing the binary code to achieve the purpose of classification and detection [6, 7]. Features extracted by manual analysis are highly accurate. However, this requires a considerable amount of manpower [8, 9]. As the malicious virus grows exponentially, the way of extracting features by manual analysis is becoming more and more expensive for this situation. Machine learning methods are highly generalized and do not require much manual work. Machine learning, because of its powerful learning ability, can learn some feature information that cannot be extracted manually. However, these methods based on machine learning are very susceptible to interference. Some existing methods, such as converting malicious code into pictures and signal frequency [2, 5], ignore the original semantics of the code and are easily interfered. As long as the malicious code author adds some byte information, or modifies the distribution of the file, the classifier can be confused. Venkatraman and Alazab [10] used the visualization of the similarity matrix to classify and detect zero-day malware. Visualization technology helps people to better understand the characteristics of malicious code, but they have not explored the application of deep learning.

In [3, 6, 11], the authors use the ASM file generated by disassembly to convert the assembly bytecode into pixel features and then use CNN to learn. Although this method takes advantage of some program information, malware authors can still make confusion by inserting external assembly instructions. Zhang et al. [12] use SVM to build a malicious code detection framework based on semi-supervised learning, which effectively solves the problem that malicious code is difficult to be marked on a large scale, and has achieved good results. There are also some methods that are based on API calls in [13]. They treat the file as a list containing only 0 or 1, with 0 and 1 representing whether or not the associated API appears. Their experiments show that the random forest classifier achieves the best result. This method mainly relies on the malicious API which could be emerged on a series of call sequence, and only the exact execution sequence can make damage on the computer system.

In [14], the authors construct behavior graphs to provide efficient information of malware behaviors using extracted API calls. The high-level features of the behavior graphs are then extracted using a neural network-Stacked AutoEncoders. On the one hand, their method of extracting behavioral graphs is very precise and helps to express the true meaning of the program fragments. On the other hand, their input vectors are constructed based on the whole sample, and the output of the model is the classification result of the whole sample. In fact, malicious fragments are only partial, which makes the malicious behavior graph easy to be overwhelmed.

Liu et al. [4] use image texture, opcode features, and API features to describe the sample files. By using the shared nearest neighbor (SNN) clustering algorithm, they obtained a good result in their dataset. Qian and Tang [15] analyze the API attributes and divide0 them into 16 categories. They propose a map color method based on categories and occurrence times for a unit time the API executed according to its categories. Then, they use the CNN model to build a classifier. Xiaofeng et al. [16] propose a new method based on information gain and removal of redundant API fragments, which effectively reduce the length of the API call sequence. The handled API call sequence is then entered into the LSTM model for training. Uppal et al. [17] use call grams and odds ratio to select the top-ranked feature segments, which are used to form feature vectors and are used to train the SVM model.

On the one hand, the above methods based on the API execution sequence are accurate, which reflect the dynamic execution information of the program. But on the other hand, due to program execution control, in a long execution sequence, the actual malicious execution code is very small or overwhelmed by a large amount of normal execution code. If the model does not learn the key malicious information, it will easily be bypassed by malicious code specifically disguised. There are also other machine learning methods to learn the features. Anderson and Roth [18] offer a public labeled benchmark dataset for training machine learning models to statically detect malicious PE files. They also construct baseline models based on gradient boosted decision tree algorithm. Even without any hyperparameter optimization, their work will still help researchers to study further in this field. In [19], the authors extract features based on the frequency of the API and compare neural networks with other traditional machine learning methods.

These methods expand the space for extracting malicious features, and improve the applicable scale of the machine learning method, and achieve good results. But they also have some limitations, mainly reflecting in the following aspects. First, manual methods have high

accuracy but require a lot of manpower, which makes them unsuitable for analyzing a large amount of malicious code. Second, machine learning is greatly influenced by the training set and its practicality is weak. For example, we have done an experiment, in which an image-based malware classifier can achieve 0.99 accuracy rate. However, after changing dataset, its performance drops sharply to about 0.73. Third, when the sample is confused, the training model is difficult to achieve good results, which reduces its practicability.

In fact, whether it is converted to images [20], signals, frequency, and other characteristics, it cannot truly express malicious code, nor does it conform to the traditional prior knowledge of malicious code. The method of extracting more efficient sequences by N-gram slicing [21, 22] only retains the sequential features of malicious code execution. The models trained with the features extracted by the common methods will have a poor effect.

Therefore, it is worth in-depth and long-term research to explore how to design a detection framework with the help of prior knowledge of malware, so that we can apply deep learning to malware detection better.

3. Our Method

From the perspective of information security, we analyze some malware and try to figure out the essential characteristics of them. After that, we confirm the local malicious characteristics of malware. Based on this, we propose a novel feature extraction method and build a detection framework based on the deep learning model.

3.1. Local Malicious Analysis. IDA Pro [23] is a powerful disassembler that can be used to disassemble binary so that by its disassembly result, we could get a sequence of API calls and analyze the behavior of the program. Through IDA Pro, we analyze a malicious code sample Trojan (VirusShare0a83777e95be86c5701aaba0d9531015 from VirusShare website [24]) as shown in Figure 1. As a result of the disassembly of IDA Pro, we can see that the Trojan's privilege operation is invoked by these six consecutive API function calls. The first function is *GetCurrentProcess*, which gets the handle of the current process by executing this function. The second function is *OpenProcessToken*, which can be used to open the current process with the handle value obtained by the first function. The third function is *LookupPrivilegeValueA*, which is used to view the permissions of the current process. The fourth function is *AdjustTokenPrivileges*, which is used to raise the permissions for the current process. The fifth function is *CloseHandle*, which closes the currently open process. The sixth function is *ExitWindowsEx*, which exits the current window. From the first function to the sixth function, this set of functions is used to elevate privilege of the current process, thus forming a malicious behavior. Because normal behavior does not take these operations to elevate its privilege, this is the difference between normal software and malware.

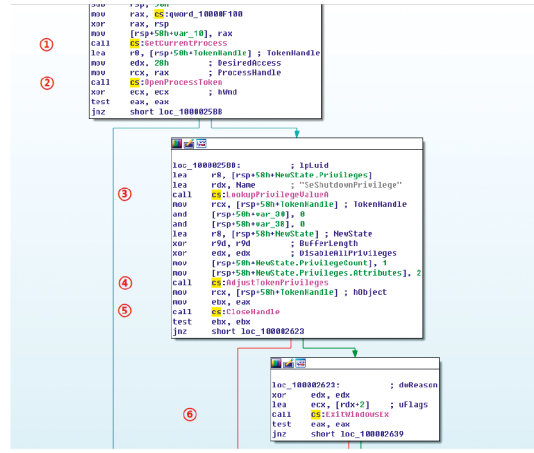


Figure 1: Trojan sample. The Figure is generated by IDA Pro. There are six different API calls marked by numbers from 1 to 6. This sequence indicates the API functions execution order.

In fact, the function of the program, whether normal or malicious, is mainly achieved through ordered API calls. Also, most of the malware is originated from benign software which is infected by malware such as viruses. From the code perspective, it is a normal code sequence that was inserted with malicious code snippets. In other words, most of the sequences of malicious code are normal, and only small segments are malicious. Malicious code also typically constructs itself by inserting normal code in this way. As shown in Figure 2, malicious code snippets are usually local and are only part of the overall code, which proves the local maliciousness of the malicious code we analyzed.

They often use the method of inserting useless APIs, conditional triggers, and confusion to counter detection. Traditional machine learning models that rely on the whole API call sequences as training data will be affected. Therefore, we propose a new extraction method based on local malicious features, and implement a framework which relies on local maliciousness to identify malicious code.

3.2. Detection System Framework. We first cut the entire API sequence into API fragments of length N. According to the local maliciousness, the API fragment of length N always retains the information of the malicious API fragment, and the API execution fragment constituting the maliciousness will always be acquired regardless of whether or not the source code is handled by confusion or deformation. Then, the tagged API fragments after de-duplication will be entered into the classifier for training, so that the classifier could detect the malicious API fragment. Furthermore, we use ensemble learning to identify malicious code by the proportion of malicious API fragments in the whole API execution sequence. These are the ideas of our entire framework.

Our framework can be divided into four parts: data processing, feature extracting, training model, and decision-making, as shown in Figure 3.

Data processing builds a word vector for all of API, and then transforms API sequences to a number list. Feature

Figure 2: The distribution of malicious code. The red parts represent malicious codes, which distribute in the whole code.

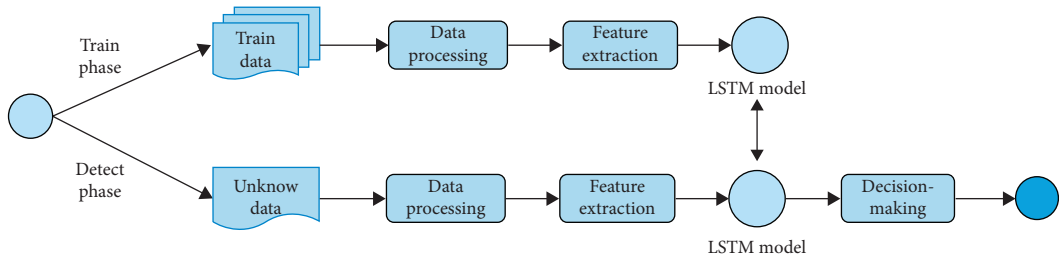


Figure 3: The framework of our method. The procedure consists of 4 steps. By data processing, the feature extracting, training model and decision-making, it gives a final classification result.

extracting uses sliding window operation to get slices with a fixed-length parameter N. We research different lengths for the sliding window to make the best result for slices. We use LSTM as our classifier to classify API fragments. Decision-making based on trained models makes a decision by voting on the basis of threshold M. Furthermore, we have a deep research on the parameters N and M. Based on this framework, our system is divided into the training phase and detecting phase, as shown in Figures 4 and 5.

Each process will be described in detail in the following sections.

(1) Data processing: in order to verify our hypothesis, we omit the process of API execution sequence extraction, which can be extracted by sandbox techniques. We use the public dataset directly, which consists of the API execution sequences that represent the executable files executing processes in the actual environment.

First, we use Word2vec [25] to build the API vocabulary by collecting all the API fragments that appear in the samples and use a number to uniquely represent an API. The samples consisting of API sequence are then replaced with a digital sequence for feature extraction, as shown in Figure 6.

(2) Feature extracting: we further analyze the malicious code and find that, in reality, in order to avoid detection, malicious code is usually inserted useless API function into the API call sequence. These malicious API calls only could be executed when the condition is triggered.

Based on the previous analysis of local maliciousness, we use the sliding window method to get the API fragment. We first set the length of the fixed sliding window to N and the step size to 1. Then, an API execution sequence with a length of M will form M − N + 1 API fragments by sliding the window. The operation diagram is shown in Figure 7.

Figure 7 above shows our sliding window approach, which is different from the N-gram method that is used in [26, 27]. The N-gram method will produce a large number of segments, and the overall position map of the different N-gram segments will not reflect the true meaning of a segment. Instead, the really useful segment information will be submerged.

After that, we label the API fragments generated by the sample with the same category tag. In fact, the API fragment is represented as a specific digital slice. By dividing the slices, these slice libraries with different labels are formed. Different from the method of information entropy adopted in [4], a de-duplication method is adopted by us to remove the same slices in the libraries by two to two.

(3) Model: the API call sequence reflects the malicious code execution order. This order is important. In the deep learning model, the RNN model has the function of remembering the sequence and is suitable for such data characteristics. Each hidden layer output of the RNN model is retained in the network so that it could combine with the next moment input of the system to determine the output of the next moment. The long and short-term memory LSTM model is an updated version of the RNN model that overcomes back-propagation gradient dispersion and gradient explosion and is more suitable for processing sequence data. Therefore, we choose LSTM as our training model. We design a three-layer LSTM as follow: each LSTM layer is connected to a dropout layer. Then, LSTM layer connects with each other. Finally, a dense layer is connected, which is used as the classifier.

The input for the LSTM model is digital slice which is formed in the feature extraction process. The model finally outputs the digital slice category.

(4) Decision-making: decision-making is designed for the purpose of classifying the unknown sample. In the training phase, we will train multiple models. In the detecting phase, because the models we trained are used to classify API fragments, we will make a decision on the unknown sample according to the digital slices which are formed by the unknown sample. Finally, classification results are determined if voting values of multiple models reach a certain

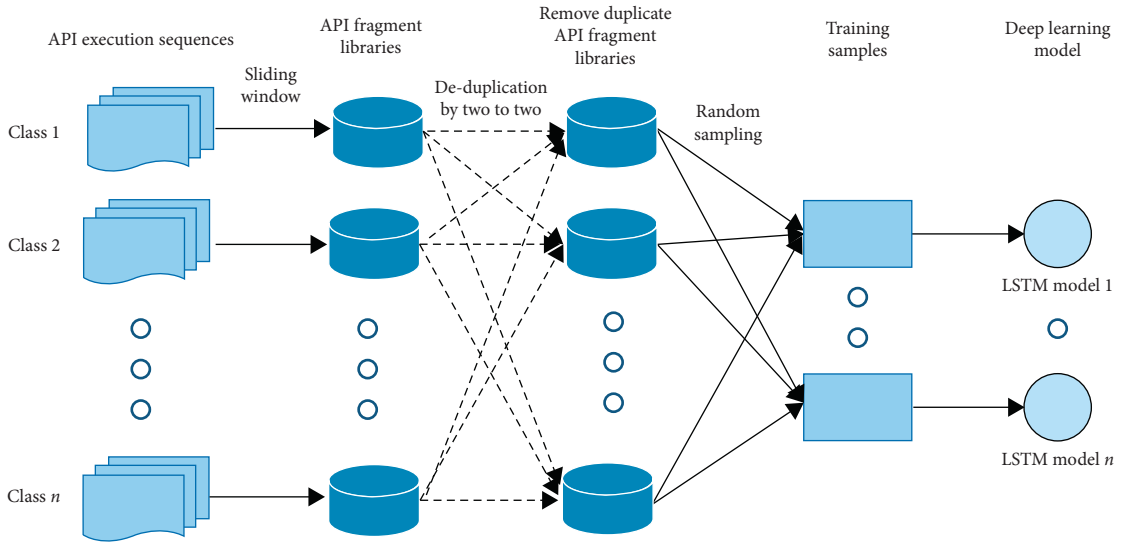


Figure 4: The training phase.

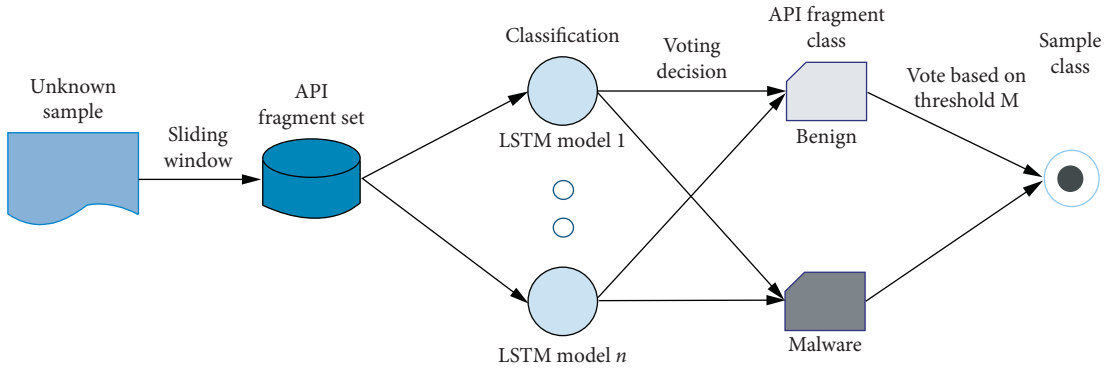


Figure 5: The detecting phase.

threshold M. We will set a threshold M and continuously optimize it during the experiment.

The details for decision-making are given as follows in Algorithm 1, which consists of SPLITWINDOW function and DECISIONMAKING function. The SPLITWINDOW function generates sets of removing duplicate API fragments. The DECISIONMAKING function makes final decision on unknown samples.

4. Experiment

4.1. Dataset. We use thedata of Alibaba-3rd-Security-Algorithm-Challenge [28] to test our method. The dataset is the API instruction sequence from the Windows executable program files in the sandbox simulation. All Windows binary executable program are desensitized. The sample data provided on this topic are collected from the Internet. Among them, the types of malicious files are the infected virus, Trojan horse program, mining program, DDoS Trojan, and extortion virus. By parsing the API data, there are a total of 12,000 samples. We randomly split the dataset into a training set and a test set. The sample distribution is shown in Figure 8.

4.2. Data Handling. In order to verify the anti-interference ability of the framework, we first randomly insert a number of invalid API functions into the API sequence for the purpose of confusion. Then, we use Word2vec to convert the API sequence into a sequence of numbers and use the sliding window to cut into the whole sequence, forming different libraries of fragments with different labels. After that, we use de-duplication method to remove redundant API fragments for each API slice library. In the training phase, because the number of malicious code files of different categories is very different, we adopt the bagging method by using random sampling with putting back to construct training sets. Then, each training set is used to train a model.

4.3. Result and Analysis. The accuracy, precision, recall, and F1-measure are selected as the evaluation indicators for the classification of samples and compared in comparison experiments, as shown below.

(1) Parameters N and M: the parameters N and M, respectively, represent the length of API slice and the threshold for judging the sample's category. We first verify the classification accuracy of API

GetSystemTimeASFileTime	1
NtAllocateVirtualMemory	11
NtFreeVirtualMemory	8
NtAllocateVirtualMemory	11
NtAllocateVirtualMemory	11
NtAllocateVirtualMemory	11
NtAllocateVirtualMemory	11
SetUnhandledExceptionFilter	13
LdrLoadD11	4
LdrLoadD11	4
LdrGetProcedureAddress	23
LdrUnLoadD11	5
NtCreateMut ant	25
NtCreateSection	41
NtMapViewof Section	78
LdrLoadD11	4
LdrGetProcedureAddress	29
LdrUnLoadD11	5
NtCreateMut ant	6
NtCreateSection	9
NtMapViewof Section	24
.....	.....
.....	.....

Figure 6: API sequence is converted to a digital sequence. Each API function corresponds to an unique number, and the entire API sequence is converted to a number for further processing.

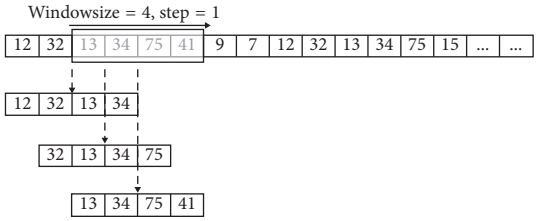


Figure 7: The operation of the sliding window. In the figure, the sliding window with window size by 4 moves rightwards with a step by 1. Each step of the window will produce a segment of length 4.

executable file for different parameters N and M. The results are shown in Figures 9–11.

As we can see from the above figures, although the length N is different, the threshold M and the accuracy are approximately subject to normal distribution. This also shows that we can definitely get an optimal value for the model. It is different from the study in [21], which determines an optimal N-gram length by the Cuckoo Algorithm [29].

Furthermore, these figures also show that, the shorter the length N, the smaller the threshold M for achieving the maximum accuracy. Conversely, the longer the length N, the larger the threshold M for achieving maximum accuracy. Furthermore, it is not difficult to infer that when the fragment length reaches the limit equal to the length of the entire sequence, the threshold at this time will be 1. It reflects the local maliciousness that we propose before, that is, the malicious fragments are only part of the entire sequence.

We then put these figures with different length parameters together, as shown in Figure 12.

It is obvious that the parameter N and M can get the best accuracy result. Also, all of the figures show the same trend that they are subject to normal distribution, which is consistent with the fact.

(2) Evaluation: we select the $N = 10$ and $M = 0.40$ as the best parameters, and evaluate different indications, that is, precision, recall, f1-score, and confusion matrix. The result is shown in Tables 1 and 2.

Table 1 shows the results of precision, recall, f1-score, and support for categories 0 and 1, respectively. Category 0 represents the normal code and category 1 represents the malicious code. It achieves high performance for classifying mission. Table 2 shows a confusion matrix for categories 0 and 1. By this, we can get the value of the true positive (TP), false positive (FP), false negative (FN), and true negative (TN). Accuracy is defined as follows:

$$\text{Accuracy (ACC)} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})}. \quad (1)$$

Finally, the accuracy is 0.9734, FPR is extremely 0, FNR is 0.0636, and the sensitivity is 0.9364. It proves that our method has an effective classification for malware.

4.4. Comparison. In order to verify the validity of our method, we design three comparative experiments. The first comparison test, without splitting the API call sequence, is directly converted into a digital sequence and sent to the same LSTM model for training. In the second comparative experiment, the same feature extraction method as the framework we proposed was used, and CNN was selected as the training model for evaluation. The CNN model is set as follows: we set up a three-layer convolution with 3, 4, and 5 as convolution kernel sizes, then it is connected to the max-pooling layer, further to a dropout layer, and finally to the fully connected layer. In the third comparative experiment, based on the same feature extraction method, SVM is selected as the training model for evaluation. The SVM model is set as follows: the penalty coefficient C is set to 10000. RBF is selected as the core, and its parameter gamma is set to 0.001, and the parameter degree is set to 3.

We choose accuracy as the evaluation criteria, and the results are shown in Figure 13. As can be seen from the results, the accuracy of our model is 0.9734, and the accuracy of using the entire API sequence is 0.8246. The accuracy of using the CNN model is 0.9325. The accuracy of using the SVM model is 0.9143.

Through comparative experiments, we can find that our method, the second, and the third methods are more accurate than the method based on the overall API execution sequence. It shows our method, that is the feature extraction method based on local maliciousness extraction API fragment, is effective. Because the sample has been confused, it also reflects that our feature extraction method has a degree of anti-interference ability. After adopting the same feature extraction method, by comparison, it can be found that the

Input: *sample*, *length* (the length of sample), *N* (the length of the window), *M* (threshold for voting), *C* (a set of all trained model for classification)
Output: *set* (store all API slices to be cut)
(1) **function** SplitWindow (*sample*, *length*, *N*)
(2) initial place in the beginning of the sample
(3) **repeat**
(4) split the sample with the solid window
(5) move the window with a step 1
(6) **until** move to the end of sample
(7) move all API slices into set
(8) Remove duplicates
(9) **return** *set*
(10) **end function**
(11)
Input: *set* (generated by Call SplitWindow ()), *M* (threshold for voting), *C* (a set of all trained model for classification)
Output: *category* (normal or malicious)
(12) **function** DECISION MAKING (*set*, *m*, *C*)
(13) **for** each $s \in$ set **do**
(14) **for** each $f \in C$ **do**
(15) $p = f(s)$
(16) **if** $p > 0.5$ **then**
(17) *s* is belong to normal slice
(18) **else**
(19) *s* is belong to malicious slice
(20) **end if**
(21) record the result for *s*
(22) **end for**
(23) **end for**
(24) $number = account(s_{malicious})$
(25) $total = account(s_{all})$
(26) **if** $number/total > m$ **then**
(27) **return** malicious
(28) **else**
(29) **return** normal
(30)

Algorithm 1: Classifying an unknown sample.

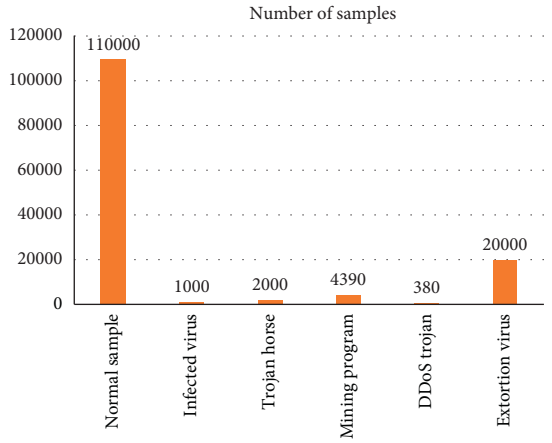


Figure 8: Data catagory numbers.

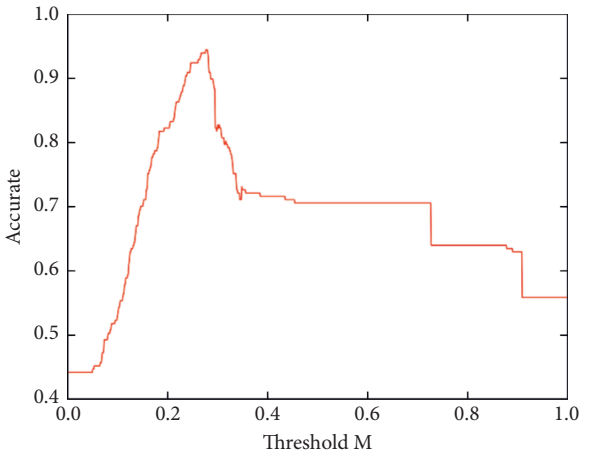


Figure 9: $N = 2$.

accuracy of the LSTM-based model is higher than that of the CNN and SVM models. Further analysis shows that our data are derived from API execution sequences with sequence characteristics. Our model is based on LSTM, which is very effective for processing data with sequential relationships. The CNN model is very effective for learning image features and is very effective for learning data with local features. SVM is a classic traditional machine learning model, but its learning ability is weaker than deep learning models such as

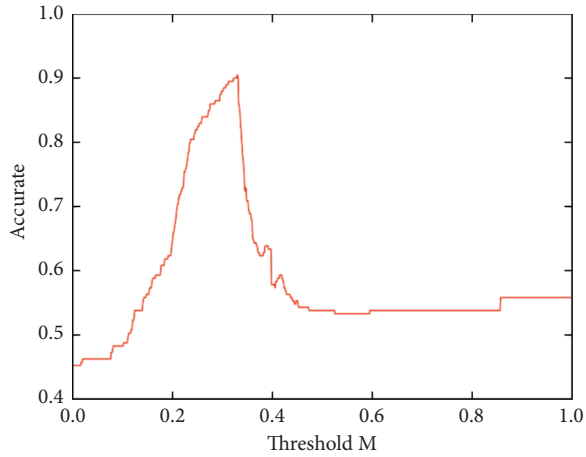


Figure 10: $N = 8$.

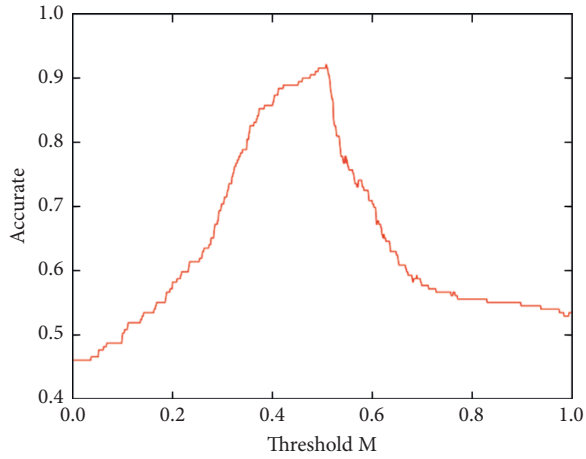


Figure 11: $N = 22$.

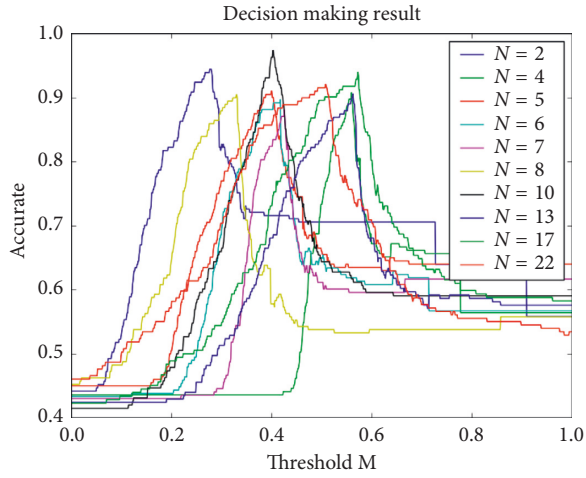


Figure 12: Different N.

LSTM and CNN. From the above analysis, we can see that our LSTM model is better for this kind of data with a sequential relationship. It is also the reason why the SVM is the lowest, CNN is the second, and our LSTM model is the highest.

Table 1: Classify report.

Classid	Precision	Recall	f1-score	Support
0	1.00	0.94	0.97	110
1	0.92	1.00	0.96	78
Avg/total	0.97	0.96	0.96	188

Table 2: Confusion matrix.

Classify	Benign	Malware
Benign	103	7
Malware	0	78

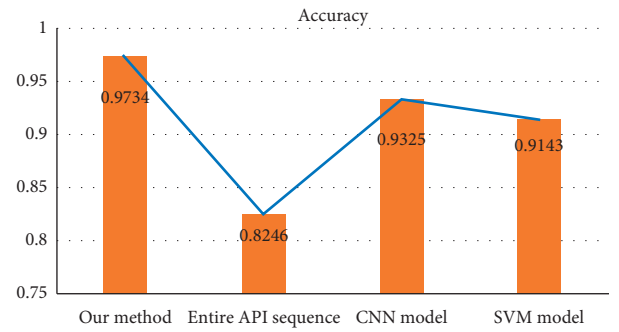


Figure 13: Comparative test.

As far as previous research is concerned, they usually did not delve into the local meaning of the API sequence. The general methods are to transform the API sequence as a whole and enter it into the model for training.

Meanwhile, the innovation of past methods about API sequence is focused on the elimination of redundant API functions or the construction of the corresponding conversion sequence [4, 15, 16]. But no matter how they reform, these ideas are still based on the overall API sequence. It makes their methods easily disturbed once malware developers take countermeasures. This inevitably leads to the lack of anti-interference for their model, which makes it difficult to have practical significance and to exert the powerful ability of deep learning.

In fact, the confused sample we constructed does not cover all the confrontation techniques, but it strongly proves the effectiveness of our extraction method based on local maliciousness and verifies our innovation.

5. Conclusion

We analyze the local maliciousness of malicious code, based on which we design a deep learning-based ensemble learning detection framework for API fragments. We use interference-handled samples for training and validation. The results show that our method can effectively resist interference. Our method also effectively explores the application of deep learning in the field of malicious code detection, which has a strong practical significance. In the future work, we will further study the combination of prior knowledge and deep learning.

Data Availability

Research data related to this article are deposited in the Ali-Tianchi contest dataset repository: Ali-Tianchi contest dataset (https://tianchi.aliyun.com/competition/entrance/231668/information).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the grants from the National Key Research and Development Program of China (Project no. 2017YFB0802800).

References

[1] K. Kosmidis and C. Kalloniatis, "Machine learning and images for malware detection and classification," in *Proceedings of the 21st Pan-Hellenic Conference on Informatics—PCI 2017*, p. 5, ACM, Larissa, Greece, September 2017.

[2] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the International Symposium on Visualization for Cyber Security*, pp. 1–7, ACM, Pittsburgh, PA, USA, July 2011.

[3] L. D. Vu Duc, *DEEPMAL: Deep Convolutional and Recurrent Neural Networks for Malware Classification*, 2018.

[4] L. Liu, B.-S. Wang, B. Yu, and Q.-X. Zhong, "Automatic malware classification and new malware detection using machine learning," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 9, pp. 1336–1347, 2017.

[5] L. Nataraj, "A signal processing approach to malware analysis," Dissertations & theses—Gradworks, University of California, Santa Barbara, CA, USA, 2015.

[6] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant, "Semantics-aware malware detection," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pp. 32–46, IEEE, Oakland, CA, USA, May 2005.

[7] C. Kruegel, W. Robertson, F. Valeur, and G. Vigna, "Static disassembly of obfuscated binaries," in *Proceedings of the USENIX Security Symposium*, vol. 13, p. 18, San Diego, CA, USA, August 2004.

[8] D. M. Chess and S. R. White, "An undetectable computer virus," in *Proceedings of the Virus Bulletin Conference*, vol. 5, pp. 1–4, Orlando, FL, USA, 2000.

[9] F. Cohen, "Computer viruses," *Computers & Security*, vol. 6, no. 1, pp. 22–35, 1987.

[10] S. Venkatraman and M. Alazab, "Use of data visualisation for zero-day malware detection," *Security and Communication Networks*, vol. 2018, Article ID 1728303, 13 pages, 2018.

[11] D. Gibert Llauradó, "Convolutional neural networks for malware classification," Thesis, Universitat Rovira I Virgili, Tarragona, Spain, 2016.

[12] K. Zhang, C. Li, Y. Wang, X. Zhu, and H. Wang, "Collaborative support vector machine for malware detection," *Procedia Computer Science*, vol. 108, pp. 1682–1691, 2017.

[13] M. Alazab, S. Venkatraman, P. Watters, and M. Alazab, "Zero-day malware detection based on supervised learning algorithms of API call signatures," in *Proceedings of the 2011 Ninth Australasian Data Mining Conference*, vol. 121, pp. 171–182, Australian Computer Society, Inc., Ballarat, Australia, December 2011.

[14] F. Xiao, Z. Lin, Y. Sun, and Y. Ma, "Malware detection based on deep learning of behavior graphs," *Mathematical Problems in Engineering*, vol. 2019, Article ID 8195395, 10 pages, 2019.

[15] Q. Qian and M. Tang, "Dynamic API call sequence visualization for malware classification," *IET Information Security*, vol. 13, no. 4, pp. 377–367, 2018.

[16] L. Xiaofeng, Z. Xiao, J. Fangshuo, Y. Shengwei, and S. Jing, "ASSCA: API based sequence and statistics features combined malware detection architecture," *Procedia Computer Science*, vol. 129, pp. 248–256, 2018.

[17] D. Uppal, R. Sinha, V. Mehra, and V. Jain, "Malware detection and classification based on extraction of API sequences," in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2337–2342, IEEE, Noida, India, September 2014.

[18] H. S. Anderson and P. Roth, "Ember: an open dataset for training static PE malware machine learning models," 2018, https://arxiv.org/abs/1804.04637.

[19] M. Alazab and S. Venkatraman, "Detecting malicious behaviour using supervised learning algorithms of the function calls," *International Journal of Electronic Security and Digital Forensics*, vol. 5, no. 2, pp. 90–109, 2013.

[20] L. Nataraj, V. Yegneswaran, P. Porras, and J. Zhang, "A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, pp. 21–30, ACM, New York, NY, USA, August 2011.

[21] G. Bala Krishna, V. Radha, and K. V. G. Rao, "ELC-PPW: ensemble learning and classification (LC) by positional patterns weights (PPW) of API calls as dynamic *N*-grams for malware perception," *International Journal of Simulation–Systems, Science & Technology*, vol. 18, no. 1, 2017.

[22] C. Liangboonprakong and O. Sornil, "Classification of malware families based on *N*-grams sequential pattern features," in *Proceedings of the 2013 8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 777–782, IEEE, Melbourne, Victoria, Australia, June 2013.

[23] IDA Pro, 2019, https://www.hex-rays.com/products/ida/.

[24] Virusshare Website, 2019, https://virusshare.com/.

[25] Word2vec, 2019, https://code.google.com/p/word2vec/.

[26] E. Menahem, A. Shabtai, L. Rokach, and Y. Elovici, "Improving malware detection by applying multi-inducer ensemble," *Computational Statistics & Data Analysis*, vol. 53, no. 4, pp. 1483–1494, 2009.

[27] Y. Ye, L. Tao, Q. Jiang, Z. Han, and L. Wan, "Intelligent file scoring system for malware detection from the gray list," in *Proceedings of the 15th ACM Sigkdd International Conference on Knowledge Discovery & Data Mining*, Paris, France, June 2009.

[28] Alitianchicontest, https://tianchi.aliyun.com/competition/introduction.htm?spm=5176.11409106.5678.1.4354684cI0fYC1&raceId=231668.

[29] Cuckoo Algorithm, 2019, https://en.wikipedia.org/wiki/Cuckoo_search.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 2798571, 9 pages
https://doi.org/10.1155/2019/2798571

Research Article

An Object Proxy-Based Dynamic Layer Replacement to Protect IoMT Applications

Bo Han,[1] **Zhao Yin-Liang**,[1] **and Zhu Chang-Peng**[2]

[1]*School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2]*School of Liangjiang Artificial Intelligence, Chongqing University of Technology, Chongqing 400054, China*

Correspondence should be addressed to Bo Han; bohan@xjtu.edu.cn

Received 24 May 2019; Accepted 26 August 2019; Published 30 September 2019

Guest Editor: Kuan Zhang

The Internet of medical things (IoMT) has become a promising paradigm, where the invaluable additional data can be collected by the ordinary medical devices when connecting to the Internet. The deep understanding of symptoms and trends can be provided to patients to manage their lives and treatments. However, due to the diversity of medical devices in IoMT, the codes of healthcare applications may be manipulated and tangled by malicious devices. In addition, the linguistic structures for layer activation in languages cause controls of layer activation to be part of program's business logic, which hinders the dynamic replacement of layers. Therefore, to solve the above critical problems in IoMT, in this paper, a new approach is firstly proposed to support the dynamic replacement of layer in IoMT applications by incorporating object proxy into virtual machine (VM). Secondly, the heap and address are used to model the object and object evolution to guarantee the feasibility of the approach. After that, we analyze the influences of field access and method invocation and evaluate the risk and safety of the application when these constraints are satisfied. Finally, we conduct the evaluations by extending Java VM to validate the effectiveness of the proposal.

1. Introduction

Internet of medical things (IoMT), which plays an important role in healthcare industry to construct the safety and healthy system in human society [1], is becoming the critical part in reducing healthcare costs, providing timely medical responses, and increasing the quality of medical treatment. IoMT, in fact, is an ecosystem of connected different kinds of sensors, including the wearable devices, medical devices, and clinical systems. Each part in the ecosystem has its own particular context, which is different from others. On one hand, various contexts should be managed by the healthcare application to provide context-based services. For example, to avoid patients' information leakage, a function component of a healthcare application is used to describe the medical histories of patients to doctor in detail, instead of collecting the sensitive information. On the other hand, the component can be applied in smart watches to regulate the patients' personal information. However, due to *crosscut*, the codes are easily scattered and tangled by context, which increase the costs of healthcare applications' development and maintenance.

Recently, some researchers have introduced the technologies from academic and industry to improve the effectiveness and performance of IoMT applications. For example, the dynamic layered-recurrent neural network was developed by Turabieh et al. [2] to recover missing data. *Data mashup*-based web technology was proposed by Elmisery et al. [3] to serve a centralized environmental monitoring service.

In addition, the programming languages are compiler-based, which means that all layers and their activation/deactivation are determined in a development stage. The contexts cannot be completely predicted at the development stage [4], healthcare applications, which limit the applications of language. Therefore, in this paper, we introduce an encapsulation mechanism-based context-orient programming paradigm (COP) in IoMT to reduce the occurrences of above-mentioned crosscuts. In COP, the application's codes are divided into different parts (i.e.,

layers), which is a new block-structured construction. Each layer is related to other contexts, where the layer activation and deactivation mechanism is provided to ensure that the codes in different layers can be executed. As a result, the application can adjust its behaviors according to context. Some extensions of Java for COP, such as JCOP [5] and EventCJ [6], have been implemented.

Virtual machine (VM), which has been considered as an intermediate layer between the executing application and the hardware [7], is widely used in big data process and analysis [8, 9], as well as increasing the possibility for dynamic code evolution. Object proxy [10] is advocated as a flexible mechanism to dynamically adapt behavior of objects to adapt the behavior of a single object. Therefore, this paper proposes a new approach, which incorporates object proxy into VM, to enhance the flexibility of applications in terms of context-based dynamical behavior adjustment. Unlike above-mentioned COP languages, based on VM, layer activation and deactivation in our approach are used to monitor the changes of context and dynamically activate or deactivate layers so that the application can adapt its behavior in context.

In the perspective of VM, a layer is a Java class. In order to support layer activation and deactivation mechanism on VM level, the object proxy is used in our approach. An object at runtime is actually an object list, which is composed of an object of a base class and objects of layer classes. The base class object is considered as a proxy object, where each method invocation is redirected to other objects in the object list. As a result, codes in the layer are executed to achieve the layer activation. Similarly, the layer deactivation is also achieved by removing or deactivating other objects in the list.

Furthermore, dynamic layer replacement can be achieved in our approach via object evolution. As an object is represented as a list, when VM monitors the changes of context, VM can evolve an object at runtime by inserting a new object of a layer class into it or deleting an object from it. Once the layer is replaced, VM removes the objects of layered classes in the removed layer from object lists and inserts new objects of layered class in the new loaded layered classes in the new layer into object lists. As a result, the application can adjust its behavior to fit the changes of contexts through dynamic layer replacement.

On the other hand, the heap and stack need to be scanned by the VM to find out all objects that to be evolved, which leads to a lot of overhead for the object evolution at runtime. To relieve the overhead, the objects are evolved lazily in our approach. Furthermore, object evolution can lead to type violation [11]. Therefore, this paper incorporates heap and address into RFT-FJ [12] to model the objects and evolutions, where the semantics of field access and method invocation on objects are described in detail. Next, the impact of the evolution on type safety is analyzed. Afterwards, the constraints on object evolution are proposed to avoid the type violation. Finally, the object evolution is presented by extending Java VM to verify our application.

The remainder of this paper is structured as follows. In Section 2, we illustrate our approach by an application. In Section 3, the related work is given. In Section 4, our approach is described. In Section 5, we present the implementation and evaluation of our approach. Conclusion is finally presented in Section 6.

2. Related Work

2.1. The Internet of Medical Things. Rodrigues et al. [13] reviewed technological advances made so far which can be used in IoMT and then analyzed the challenges to be overcome. Ullah et al. [14] presented a new semantic model for patients' e-health based on existing technologies, which are used to deploy IoT in the field of medical and smart health care. Turabieh et al. [2] introduced a dynamic L-RNN from artificial intelligence to recover missing data from IoMT applications to guarantee high quality of services to the end users. Kernec et al. [15] presented place of radar for assisted living in the context of IoT for health and pointed out how important the context is. Elmisery et al. [3] used data mashup from web technology to serve a centralized environmental monitoring service. Haoyu et al. [16] used cloud computing to provide real-time sleep apnea detection. They both introduced technologies from other research areas to IoMT. Similarly, we introduce COP programming paradigm from programming language area to reduce codes' scatter and tangle in IoMT applications. Joyia et al. [1] pointed out some challenges in IoMT, where software implementation is still full of challenges. We believe our work makes an important step toward resolving these challenges.

2.2. Object Evolution. In order to support context awareness, object composition has been used to represent objects with some dynamic behaviors [7, 17].

An object evolution approach is proposed in [7]. The approach is based on composition and delegation while our approach is based on composition and proxy. This is the most important difference between them. Compared with delegation, *proxy* is easier to achieve semantic analysis and implementation of layer activation and deactivation, although it suffers from a little less flexibility.

In [18], an evolution approach based on version consistency is proposed, which is similar to our evolution approach. However, its version consistency is applied for the evolution of components in distributed system while ours is applied for objects. This is the most important difference between the two approaches in terms of version consistency.

In [17], object composition and delegation with a static type discipline are also used for dynamic behavior adaptation of objects. However, the object in [17] is only explicitly composed while the composition is implicit in our approach. Furthermore, object composition in [17] is used for dynamic inheritance and code reuse while object composition in our approach is used for dynamic behavior adaptation of the program.

Granule-oriented programming [19] uses *fitness* to determine which components are available for applications. In [12], *predicate* is applied to determine the applicability of

context-specific methods. Our approach also uses *predicate* for the similar purpose. However, there are some differences between them. *Predicate* in [19] is not formally defined while *fitness* in [12] and our approach is formally defined as a predicate method. Moreover, *fitness* in [12] is applied to select context-specific methods of composed classes while *predicate* in our approach is applied to select context-specific methods of composed objects.

In [20], various problems of software evolution and their corresponding solutions are presented and classified into several types. Some required operations on evolution are formally described and analyzed to ensure the consistency before and after evolution.

2.3. Context-Oriented Programming. In [6], an approach that incorporates features of AOP is presented to implement features of COP. Both it and our approach share some common features. Firstly, the predicate is used to determine the applicability of the context-specific method. Secondly, layer activation is on a per-instance basis rather than a per-thread basis. However, our approach is VM-based, while the approach presented in [6] is compiler-based. Although the approach incorporates features of AOP, dynamic aspect weaving is not incorporated. Therefore, the approach does not support dynamic layer addition while our approach supports it. This is a significant difference between the two approaches.

ContextFJ [21] is an extension of Featherweight Java, which models language mechanisms for context-oriented programming. Although ContextFJ and our approach both directly express context-dependent behavior, there are still some differences between them. Firstly, ContextFJ and our approach use block-structured constructs and object proxy for dynamic behavior adaptation, respectively. Secondly, ContextFJ does not allow layers to introduce new methods, but our approach does.

3. System Model

In this section, we first describe problems caused by Java-based COP languages by a simple healthcare application and then propose object evolution approach, a VM-centric approach, to solve them.

A healthcare application code snippet written in a Java-based COP language is shown in Figure 1, which is used to display healthcare schedule information for users. If the current device is a doctor's computer, the application can show more details about medical histories of their patients and must not show patients' sensitive personal information, protecting their personal privacy. On the other hand, if running on patients' smart watches, the application can briefly show important medical schedule, including some personal information.

The basic part of the application is a Java program, and classes in the part are called basic ones, not considering runtime context. The rest are two layers L1 and L2. A layer is defined by the keyword *layer*, consisting of a predicator, and a Java class, called layered class. The predictor is used to check whether or not the layer matches current execution context. If so, the methods in the layered class can be executed.

```
class Device{
  public Static String device;
  Device(String d){
     device = d;
  }
}

class Display{
   public void displayCalendarItem(){…}
}

public class test{
  public static void main(String[] args){
     Display dis = new Display();
     dis.displayCalendarItem();
  }
}

Layer L1(Device){
   boolean predicate(){return Device.device.equal("Monitor"); }
   class Device{
     public void displayCalendarItem(){…}
   }
}

Layer L2(Device){
   boolean predicate(){return Device.device.equal("watch"); }
   class Device{
     public void displayCalendarItem(){…}
   }
}
```

Figure 1: Dynamic layer replacement.

In compiler-based COP languages, layer activation and deactivation are part of codes, which means if some layers no longer match the current context, an application has to rewrite, recompile, and re-execute. This not only shortens its life cycle but also makes it not adjust its behaviors to the context well.

For solving such problems, our approach proposes that VM is responsible for layer activation and deactivation. That is, each time methods in basic classes are invoked, VM is responsible for selecting and executing methods in proper layers, which match the current execution context. As a result, each time a layer has bugs or no longer matches new execution context, a new layer can easily be dynamically loaded by VM to replace the old layer.

To achieve our approach, the object structure is extended, as shown in Figure 2. An object at runtime is represented into a double linked list. The first element of the liked list is an object of a based class. The other elements are objects of layer classes. For example, when the layer L1 is activated, a runtime object O is composed of O_b and O_{L1}, an object of the base class calendar and that of the layered class, respectively. VM redirects all method invocations on O_b to these O_{L1} as long as L1 matches the current execution environment. Actually, O_b is considered as an object proxy.

On the other hand, if L1 is replaced with L1′, VM evolves O to O′ by replacing O_{l1} with $O_{l1'}$, a new object created by VM. Since only O_b points to O_{L1}, the redirection can avoid scan heap and stack.

Furthermore, if there are many objects containing objects of L1, they must be not immediately evolved because they may be accessed very soon. Therefore, our approach evolves these objects in a lazy manner, that is, an object is

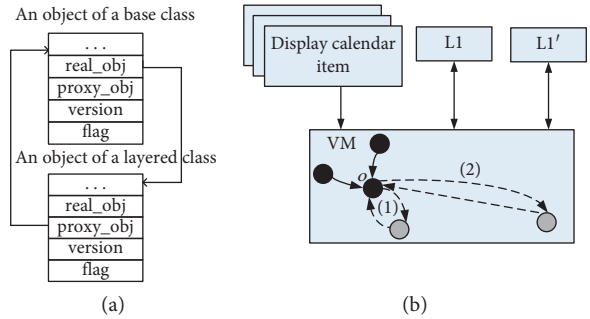


Figure 2: Object list. (a) Proxy-based object structure. (b) Object evolution.

evolved only when it is accessed. To check whether or not an object is evolved, a version number is added to structures of a class and an object. The version numbers of a classes and an object are initialized when they are loaded and created, respectively. Specifically, when a layer is replaced, due to context mismatch, the version number of a base class is increased by 1 only if the same name layered class in the layer is loaded. And the version number of the layered class is initialized with that of the base class. When an object is created, its version number is initialized with that of its class. If version numbers of a class and an object are different, then the object will be evolved when it is accessed, and version numbers of all elements in the object are updated. As a result, our approach may avoid scan heap and stack to evolve objects and improve object evolution efficiency.

Our object evolution approach is summarized as follows:

(1) For an object O represented as $O_b :: O_{CL}$, if there is a method invocation on it, the version numbers of O_b are firstly compared with those of its base class C.

(2) If equal, the object O is not evolved and the invocation will be executed. Specifically, if the field "real_obj" in O_b points to null, which means O_{CL} is null, the method will be resolved in C and executed. Otherwise, the method will resolve the object C_L pointed by the real_obj field in O_b. As a result, methods which defined layered classes are executed

(3) If not equal, which means at least a layer replacement has occurred, then the object O has to be evolved by VM. Specifically,

   (i) If O_{CL} is null, then an object of a layered class named C is firstly created, and composed a double linked list with O_b. Furthermore, the version number of O_b will be updated. Finally, VM executes step 1 again.

   (ii) If O_{CL} is not null, then O_{CL} will be replaced with a new object O'_{CL} of a new layered class C from a new layer. This is achieved by redirecting the real_obj field in O_b to the newly created object O'_{CL} and the field proxy_obj in O'_{CL} to O_b. Figure 2(b) shows the evolution process. Furthermore, the version number of O_b will be updated. Finally, VM executes step 1 again.

At runtime, objects may be replaced, according to Step 3, which results in type violation. In the next section, we model objects and their evolutions and then formally describe field access and method invocation on objects. Afterwards, some constraints are proposed to ensure that object evolution does not cause type violation.

4. Object Evolution

We first formally describe classes, objects and field access, and method invocation on objects. Afterwards, we analyze the impact of objects' evolutions on them and make some constraints to ensure such evolutions do not result in type violation. In this paper, classes are defined as sets and objects are formally described as a set of addresses in a heap.

Definition 1 (class definition). A class C can be represented by a set $\{\overline{Tf}, \overline{T \longrightarrow T'm}\}$ composed of fields and methods, where $\overline{Tf}$ is the abbreviation of $T_1 f_1, \ldots, T_n f_n$, and T and f denote a type and a field name, respectively. $\overline{T \longrightarrow T'm}$ is similar.

Two auxiliary functions S_f and S_m return the field set and the method set of a specific class C, whose definitions are given as follows:

$$S_f(C) = \begin{cases} \varnothing, & \text{if } C = \varnothing, \\ \{\overline{f}\}, & \text{if } C = \{\overline{Tf}, \overline{T \longrightarrow T'm}\}, \end{cases}$$
$$S_m(C) = \begin{cases} \varnothing, & \text{if } C = \varnothing, \\ \{\overline{m}\}, & \text{if } C = \{\overline{Tf}, \overline{T \longrightarrow T'm}\}. \end{cases} \quad (1)$$

Definition 2 (combination class). $C :: C_L$ is a combination of two same name classes, where C and C_L denote a base class C and a layered class C, respectively. The combination contains all elements of C and C_L. $O_b :: O_L$ is an object of $C :: C_L$, which can be described as the two-tuple $\langle C :: C_L, \overline{f} : \overline{l} \rangle$ and $\overline{f} = S_f(C) \cup S_f(C_L)$.

Definition 3 (object definition). An object is represented as a two-tuple $\langle C, \overline{f} : \overline{l} \rangle$ or $\langle C :: C_L, \overline{f} : \overline{l} \rangle$. The first element is the type of the object, and the second one is its fields and addresses of these fields, where l denotes an address.

Based on the above definitions, we formally describe field access and method invocation on objects by a computation state. A computation state can be described as the pair of an expression and heap, $(e, \mathbb{H})$, where an expression may be a field access $l.f$ or a method invocation $l.m(\bar{l})$. Therefore, the reduction relationship can be described by $e, \mathbb{H} \longrightarrow e', \mathbb{H}$, which indicates that a computation state $e, \mathbb{H}$ can be reduced to another state $e', \mathbb{H}$. Furthermore, $\longrightarrow^*$ denotes the reflexive transitive closure of $\longrightarrow$.

The rule for field access (E-PROJ$_1$ and E-PROJ$_2$): for an object, if its address in the heap $\mathbb{H}$ is l, its type is C or C :: C_L, and the field f has been defined, then the field access can be reduced to its value.

The rule for the method invocation (E-LINVK, E-LINVK$_2$, E-LINVK$_3$, and E-CINVK): compared with field access, the method invocations are more complex, since they contain context match and object evolution. Their main ideas are (1) if the receiver of a method invocation is only an object of the base class C, the invocation is directly reduced to its method body, where parameters are replaced by arguments, and this is replaced by the receiver and (2) if the receiver of a method invocation is an object of a combination class C :: C_L, the method invocation will be redirected to the same type method in a layered class C_L. If the layer which contains C_L matches the current context, then the invocation can be reduced to the method body of the method. Otherwise, the object has to be evolved. Specifically, rules for method invocation are described as follows:

(i) E-INVK rule: for an object O represented as O_b :: O_{CL}, its address in the heap $\mathbb{H}$ is l, and its type is a base class C. If the method m is resolved in C, then the invocation is directly reduced to its method body, where parameters $\bar{x}$ are replaced by arguments $\bar{l}$, and this is replaced by l.

(ii) E-LINVK rule: for an object O represented as O_b :: O_{CL}, its address in the heap $\mathbb{H}$ is l, and its type is a combination class C :: C_L. If a method m is resolved in C_L and the layer containing C_L matches the current context, then the invocation can be reduced to the method body in C_L, where parameters $\bar{x}$ are replaced by arguments $\bar{l}$, and this is replaced by l.

(iii) E-LINVK$_2$ rule: for an object O represented as O_b :: O_{CL}, its address in the heap $\mathbb{H}$ is l, and its type is a combination class C :: C_L. If a method m is resolved in C_L but the layer does not match the current context, and a proper layer, which matches the current context, cannot be found out or provided, then the invocation can be reduced to the method body of the same type method in the base class C, where parameters $\bar{x}$ are replaced by arguments $\bar{l}$, and this is replaced by l.

(iv) E-LINVK$_3$ rule: for an object O represented as O_b :: O_{CL}, its address in the heap $\mathbb{H}$ is l, and its type is a combination class C :: C_L. If a method m is resolved in C_L but the layer does not match the current and a proper layer can be found out or provided, then the object is firstly evolved by VM, that is, an new object of $C_{L'}$ is created and replaces the object of C_L in O, and then the object's type is also updated to C :: $C_{L'}$. Afterwards, the invocation is reduced to the method body of the same type method in $C_{L'}$, where parameters $\bar{x}$ are replaced by arguments $\bar{l}$, and this is replaced by l.

(v) E-CINVK rule: for an object O represented as O_b :: O_{CL}, its address in the heap $\mathbb{H}$ is l, and its type is a combination class C :: C_L. If a method m is invoked by another method in the same layer class, then the context match is not checked and the invocation is reduced to its body, where parameters $\bar{x}$ are replaced by arguments $\bar{l}$, and this is replaced by l.

In the above-mentioned rules, the function *mbdoy* () models the semantic of the method resolution in Java. It returns a method body of a method in a class. More details about it are shown in [7, 12]. "predicate() true IN C_L" represents the layer L that matches the current context while "predicate() false IN C_L" represents the layer L that does not match.

The notation $\Longrightarrow$ in the E-LINVK$_3$ rule denotes object evolution. Object evolution may result in the type violation when evolved object is being accessed. For example, if E-LINVK$_1$ is interrupted after it executes for a moment, then E-LINVK$_3$ is executed and an object l is evolved. When E-LINVK$_1$ is executed again, the method may not exist, due to l's change, yielding the type violation. Therefore, some constraints should be proposed for object evolution so that the type violation can be avoided.

Next, we firstly formally describe object evolution and present some constraints on it, which are shown in the following definition.

Definition 4 (valid object evolution). Assume that an object l is evolved to $\underline{l}$, denoted by $l \Longrightarrow \underline{l}$; since a layered class C_L is replaced by another well-formed $C_{L'}$, we have $\mathbb{H}(l) = \langle C :: C_{L1}, \overrightarrow{f} : \overrightarrow{l} \rangle$, $\mathbb{H}(\underline{l}) = \langle C :: C_{L2}, \overrightarrow{f}' : \overrightarrow{l}' \rangle$, $\overrightarrow{f} = S_f(C) \cap S_f(C_{L1})$, $\overrightarrow{f}' = S_f(C) \cup S_f(C_{L2})$. Furthermore, the evolution is valid only if the following conditions hold:

(1) $\forall m$ m is in stack, if $m \in l$, then $m \in \underline{l}$

(2) $\forall f$ $f \in S_f(C_L)$, f is only accessed by $m \in S_m(C_L)$

(3) $S_m(C_{L1}) \subseteq S_m(C_{L2})$, and $\forall m$ $m \in S_m(C_{L1})$ then m $type(m, C_{L2}) <: mtype(m, C_{L1})$

The first constraint requires that object evolution cannot have influence on running methods. This ensures these methods still exist after object evolution, avoiding runtime errors. The second constraint ensures fields in a layered class are only accessed by methods in the class, which ensures accessing fields do not lose after object evolution. The function *mtype* [7, 12] returns the type of a method in a class. The third constraint requires C_{L2} to have the same methods as C_{L1}, which ensures there are no dynamic-type violations after objects' evolution.

Based on the above definition, if base classes and layer classes are well typed, the following theorem can ensure type violation does not occur after object evolution. We denote expressions before and after objection evolution by e and $\underline{e}$, respectively.

Theorem 1. *Assume that e contains l and that $l \Longrightarrow \underline{l}$ results in $e \Longrightarrow \underline{e}$; if e has the property of type-safe and all layer class C_L in each replacement are well typed, then $\underline{e}$ has the property of type-safe too.*

The theorem intuitively indicates that if any component of a well-typed expression e is replaced with another well-typed component, then e is still well typed. This is very similar to industrial products in the real world. For example, if a component of a qualified car is replaced with another qualified one, then the car is still qualified.

Proof. We prove the theorem by induction.

(1) Basic: if l is unchanged, then e is unchanged. According to the assumption, the theorem holds.

(2) Reduction: assume that $e \Longrightarrow \underline{e}$ results from $l \Longrightarrow \underline{l}$, which is caused by C_{L2} replacing C_{L1}. If e is type-safe, $\underline{e}$ is type-safe.

Proof by Contradiction. If $\underline{e}$ has type violation caused by $l \Longrightarrow \underline{l}$. Without loss of generality, assume that $\mathbb{H}(l) = \langle C :: C_{L1}, \overline{f} : \overline{l} \rangle$, $\mathbb{H}(\underline{l}) = \langle C :: C_{L1}, \overline{f}' : \overline{l}' \rangle$:

(i) If $\underline{l}$ contains a static-type violation, according to the static-type violation definition [11], we have $\underline{l}.f$ and $f \in \overline{f} \wedge f \notin \overline{f}'$ or $\underline{l}.m(\overline{l})$ and $m \in S_m(C) \cup S_m(C_{L1}) \wedge m \notin S_m(C) \cup S_m(C_{L2})$.

(a) Firstly, if there exists $\underline{l}.m(\overline{l})$ and the method m is being executed, then according to the first constraint of valid object evolution, m is not in the stack, e.g. $\underline{l}.m(\overline{l})$ is not being executed, which is contradicted with the assumption.

(b) Secondly, if there exists $\underline{l}.m(\overline{l})$ and m is not being invoked, then according to the condition that e and C_L are well typed, only base method invocations can be redirected to the corresponding methods of layered classes. Therefore, we have $m \in S_m(C)$, which contradicts with the assumption.

(c) If $\underline{l}'.f$ and $f \in \overline{f} \wedge f \notin \overline{f}'$, then $f \in S_f(C_{L1}) \wedge m \notin S_f(C_{L2})$. According to the second constraint of valid object evolution, fields in a layered class only are accessed only by methods in the same class. Because m is not in the stack, the field f is not accessed. The result is contradicted with the assumption. Finally, the theorem holds.

$$
\begin{aligned}
&\text{E-PROJ}_1 \quad l.f_i, \mathbb{H} \longrightarrow l_i, \text{ if } \mathbb{H}(l) = \langle C, \overline{f} : \overline{l} \rangle \\
&\text{E-PROJ}_2 \quad l.f_i, \mathbb{H} \longrightarrow l_i, \text{ if } \mathbb{H}(l) = \langle C :: C_L, \overline{f} : \overline{l} \rangle \\
&\text{E-INVK} \quad l.m(\overline{l}), \mathbb{H} \longrightarrow [\text{this} \longmapsto l, \overline{x} \longmapsto \overline{l}]e, \mathbb{H} \text{ if } \mathbb{H}(l) = \langle C, \overline{f} : \overline{l} \rangle \text{ and } mbody(m, C) = (\overline{x}, e) \\
&\text{E-CINVK} \quad l.m(\overline{l}), \mathbb{H} \longrightarrow [\text{this} \longmapsto l, \overline{x} \longmapsto \overline{l}]e, \mathbb{H} \text{ if } \mathbb{H}(l) = \langle C :: C_L, \overline{f} : \overline{l} \rangle \text{ and } mbody(m, C :: C_L) = (\overline{x}, e) \\
&\text{E-LINVK} \quad l.m(\overline{l}), \mathbb{H} \longrightarrow [\text{this} \longmapsto l, \overline{x} \longmapsto \overline{l}]e, \text{ if } \mathbb{H}(l) = \langle C :: C_L, \overline{f} : \overline{l} \rangle, mbody(m, C :: C_L) = (\overline{x}, e), \\
&\qquad \text{predicate ( ) true IN } C_L \\
&\text{E-LINVK}_2 \quad l.m(\overline{l}), \mathbb{H} \longrightarrow [\text{this} \longmapsto l, \overline{x} \longmapsto \overline{l}]e, \mathbb{H} \text{ if } \mathbb{H}(l) = \langle C :: C_L, \overline{f} : \overline{l} \rangle, \text{predicate ( ) false IN } C_L, \\
&\qquad (\exists C_{L'} : \text{predicate ( ) true IN } C_{L'} \wedge m \in C_L^{\wedge}), mbody(m, C) = (\overline{x}, e) \\
&\text{E} - \text{LINVK}_3 \quad l.m(\overline{l}), \mathbb{H} \longrightarrow l.m(\overline{l}), \left[l \Longrightarrow \langle C :: C_L', \overline{f'} : \overline{l} \rangle \right] \mathbb{H}, \text{where } \overline{f'} = S_f(C) \cup S_f(C_{L'}) \text{ if } \mathbb{H}(l) = \langle C :: C_L, \overline{f} : \overline{l} \rangle, \\
&\qquad \text{predicate ( ) false IN } C_L, \exists C_{L'} : \text{predicate ( ) true IN } C_{L'} \wedge m \in C_{L'}, safe( ).
\end{aligned}
\tag{2}
$$

(ii) If $\underline{l}$ contains a dynamic-type violation, then we have $C :: C_{L1} \not< : C :: C_{L1}$. Assume that l's type is $C :: C_{L1}$ and $\underline{l}$'s type is $C :: C_{L2}$. According to the third constraint of valid object evolution and the subclass relationship record, we have $C :: C_{L1} < : C :: C_{L2}$, which is contradicted with the assumption. So, the dynamic-type violation is nonexistent. The theorem holds. □

5. Implementation and Evaluation

In this section, we briefly describe how to extend a Java virtual machine to implement runtime object evolution, a VM-centric approach.

5.1. Implementation. JamVM is a compact Java virtual machine (JVM) which conforms to JVM specification version 2. In comparison to most other VMs, JamVM is extremely small; hence, it is easier for us to extend. In order to implement object evolution, we mainly extend JamVM's class loader, interpreter, resolution, and memory allocation.

JamVM represents objects with the structure *object*. In order to support object evolution based on object proxy, we added two pointers real_obj and proxt_obj into the structure. The extended structure is shown in Figure 2(a). Each time an object is created, these pointers are initialized to null. At runtime, they are used to create a double-linked object list, a runtime object. Furthermore, a variable of integer type, *flag*, is added into the structure. If its value is zero, VM does not check whether or not a layer matches the current context. Otherwise, the check is executed.

In our approach, objects are lazily evolved. As mentioned above in Section 3, lazy evolution is implemented by comparisons of version numbers of a class and an object. Therefore, we also added a variable of integer type, *version*, into the *object* and *classblock* which is a structure to represent as loaded classes. Afterwards, we extend the class loader so that *version* may correctly initialize each time a class is loaded. In general, each time a base class is loaded, its *version*'s value is initialized with 0. Each time a layered class is loaded or removed due to a layer replacement, its *version*'s value is increased by 1. On the other hand, each time a layered class is loaded, its *version*'s value is initialized with that of a base class with the same name. We also extend the new instruction so that when an object is created, its *version*'s value is initialized with the *version*'s value of its class.

The interpreter is responsible for dealing with method invocation. The normal execution process of method invocation is illustrated in the left of Figure 3. Each time dealing with an instruction for method invocation, the resolution module firstly resolves the invoked method and then the execution module executes the resolved method. However, if the receiver is an object list, the process should be changed, as shown in Figure 3. Particularly, a compare module and evolution module is added into method invocation process.

We extend semantics, three instructions in the interpreter module of JamVM to achieve the change. The three instructions are invokevirtual, invokespecial, and invokestatic, respectively, which are responsible for doing with method invocations in different conditions.

(i) If proxy_obj and real_obj of the receiver of the invocation are null, the receiver is only an object of a base class. Firstly, version numbers of the object and its class are compared. If equal, the method resolution just looks for the invoked method in the based class and executes its body. This is consistent with the semantic of E-INVK. Otherwise, the object is evolved. If not equal, VM evolves the object. If evolution succeeds, context match is checked. If it matches, then the resolution module looks for its method body in real_obj and executes it. Otherwise, the module looks for its method body in the object and executes it.

(ii) For the receiver, if its proxy_obj is null but its real_obj is not null, which means that the receiver is an object list containing at least two objects, the method invocation occurs on the first object of the list, an object of a base class. In this case, version numbers of the first object in the list and its class need to be compared. If they are not equal, the receiver is evolved by VM. If the evolution succeeds and context match is checked, then the resolution module looks for its method body in real_obj and executes it. Otherwise, the module looks for its method body in the object and executes it.

(iii) For the receiver, if its a proxy_obj, the method invocation occurs on an object of layered class. In this case, version numbers of the first object in the list and its class need to be compared. If they are not equal, the receiver is evolved by VM. If the evolution succeeds, then the resolution module looks for its method body in real_obj and executes it. Otherwise, the module looks for its method body in the object and executes it.

For object evolution by triggering layer replacement, the interpreter searches a specific directory to look for a proper layer which holds the following conditions:

(i) Matches the current context

(ii) Contains a layer class, which has the same type method with the current invoked method

Once VM finds out a proper layer, layered classes in it are loaded and new objects are created to evolve the receiver of the current method invocation.

In this section, our implementation proves that our approach is feasible. In the future work, we plan to support object evolution in a multithread context.

5.2. Evaluation. In this section, we discuss our runtime measurements. For illustration purposes, methods in classes and layers are called as plain methods and layered methods, respectively. We set up micromeasurements to assess the performance of the execution of the layered method. Our measurements were all run on 2.2 GHz Intel Core2Duo with 3 GB of RAM, running Ubuntu 10.04 LTS.

Java Grande benchmark framework [14] contains all kinds of method invocations and object creations and access in Java to evaluate performance of a Java VM, for example, class/instance method invocations and class/instance object creations. To evaluate our approach, we add at least two layers for each kind of method invocations and object creations and access in Java Grande benchmark framework, respectively. One of the two layers fits the current execution

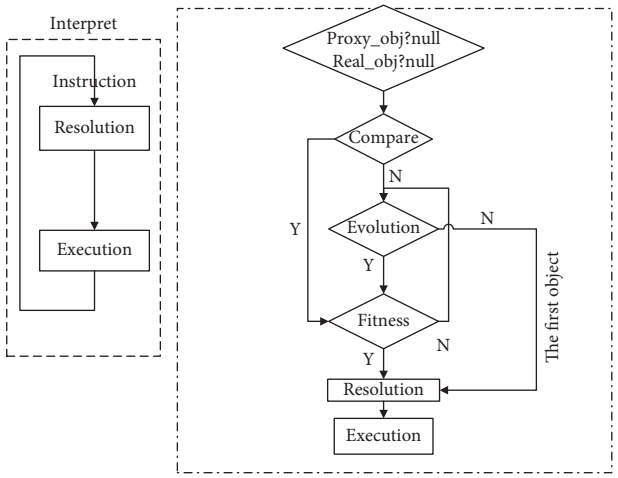


Figure 3: Method execution process.

environment while the others do not fit. On the other hand, both JamVM and our extension to JamVM run on Ubuntu 10.04 LTS operation system with Java 1.6 version.

Based on our experiment environment, we first evaluate the performance of the execution of the layered method and then compare the execution performance of method invocations on an extension to JamVM and on standard JamVM. The benchmark includes synchronized and nonsynchronized plain instance methods that calculate the power of a given number. The main method in the benchmark contains four types of plain method calls: synchronized and nonsynchronized instance methods, and calls to synchronized and nonsynchronized class methods. Each type of method call is executed 10^9 times.

5.2.1. Performance of Layer Replacement Based on Object Evolution. After 17 times execution of benchmark, the results are shown in Figure 4. The results show that the layer replacement efficiency caused by class method invocation is 468/ms. The layer replacement efficiency caused by instance method invocation is 414/ms. The difference between the two efficiencies are caused by the following reasons. Firstly, class method invocation has nothing to do with object evolution; hence, new object creations and object list constitutions are unnecessary. Moreover, compared to the synchronized method, the influence of an asynchronized method on the layer replacement is smaller. The reason is because execution of unlocking the asynchronized method needs more time.

5.2.2. Performance of Method Call Based on Object Evolution. The layer replacement based on object evolution is implemented by the extension of a virtual machine, which could impact the execution performance of method invocations on objects. In order to evaluate the influence, we compare the execution performance of method invocations on the standard JamVM and its extension by running benchmark 17 times on them, respectively.

The experiment results are shown in Figure 5. The results show that the execution performance of method invocation

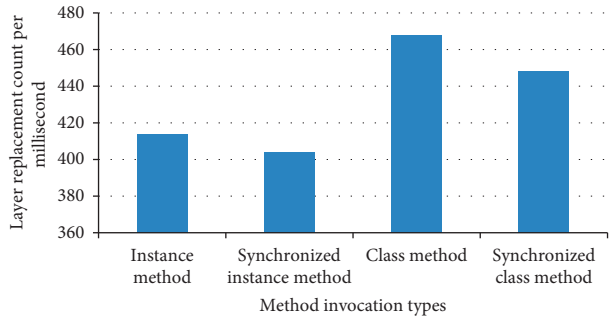


Figure 4: Layer replacement performance.

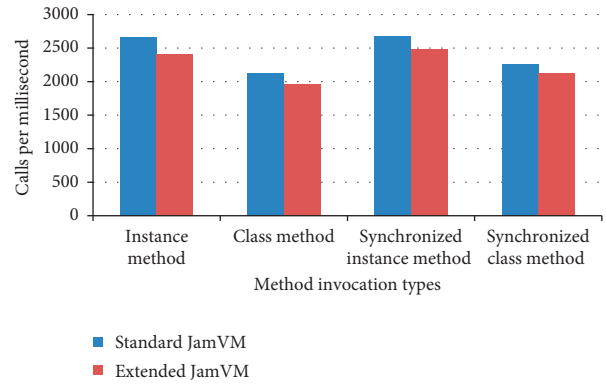


Figure 5: Method execution performance.

on the extended virtual machine is from 6% to 10% lower than that on the standard JamVM. Particularly, the execution performance of instance method invocations on the extension is close to 90% of that on standard JamVM. The execution performance of synchronization class method invocations on the extension is close to 94% of that on standard JamVM. The main reason for the low performance is that the extension JamVM has to check whether or not the receiver of a method invocation contains objects of layered classes. Furthermore, it needs to check whether or not the invoked method matches the current context.

5.3. Discussion. Our approach supports layer replacement by extending *object* represented in Java VM and semantics of method resolution, which gives rise to overhead. In space complexity, an object in our approach may cost extra tens of bytes because it contains an object list. In time complexity, as mentioned above in *Implementation* section, our approach usually does two extra operations. The first one is to locate an object in an object list, and the second one is to execute the *fitness* method. They are the root cause to overhead, which ranges from 6% to 10% in our experiment. In the future, we intend to introduce a cache mechanism to reduce an object in size and avoid *fitness* method execution as soon as possible.

6. Conclusion

Healthcare applications have to run on various medical devices, which have their own particular contexts. These

contexts crosscut healthcare applications' codes, making them more scattered and tangled and increasing costs of their development and maintenance. Some COP languages address *crosscut concern*; however, they have some deficits that new layers cannot be loaded at runtime to replace layers which no longer match the current context.

In this paper, we propose a new approach to support context-based dynamic behavior adjustment of applications. Unlike existing COP languages, our approach uses object proxy in the VM level to support layer activation and deactivation. Furthermore, as our approach is based on VM, it more easily leverages VM to dynamically replace layers so that applications can better adapt new context. In order to verify our approach, we first formally describe it and prove it, ensuring type-safety properties of applications only if some constraints are held. Finally, we present how to implement our approach by extending Java VM.

Data Availability

The experiment platform (i.e., extended JamVM) and results used to support the findings of this study have been deposited in https://github.com/zcp/jvm_extension.git, and they 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 was supported in part by a grant from NSFC under Grant nos. 61571286 and 61640219.

References

[1] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman, "Internet of medical things (IOMT): applications, benefits and future challenges in healthcare domain," *Journal of Communications*, vol. 12, no. 4, pp. 240–247, 2017.

[2] H. Turabieh, A. Abu Salem, and N. Abu-El-Rub, "Dynamic L-RNN recovery of missing data in IoMT applications," *Future Generation Computer Systems*, vol. 89, pp. 575–583, 2018.

[3] A. M. Elmisery, M. Sertovic, and B. B. Gupta, "Cognitive privacy middleware for deep learning mashup in environmental Iox," *IEEE Access*, vol. 6, pp. 8029–8041, 2018.

[4] P. Geiger, M. Schickler, R. Pryss, J. Schobel, and M. Reichert, "Location-based mobile augmented reality applications: challenges, examples, lessons learned," in *Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST 2014)*, pp. 383–394, Special Session on Business Apps, Barcelona, Spain, April 2014.

[5] M. Appeltauer, R. Hirschfeld, H. Masuhara, M. Haupt, and K. Kawauchi, "Event-specific software composition in context-oriented programming," in *Software Composition*, pp. 50–65, Springer, Berlin, Germany, 2010.

[6] T. Kamina, T. Aotani, and H. Masuhara, "EventCJ: a context-oriented programming language with declarative event-based context transition," in *Proceedings of the Tenth International Conference on Aspect-Oriented Software Development—AOSD'11*, Porto de Galinhas, Brazil, March 2011.

[7] C. Zhu, Y. Zhao, B. Han, Q. Zeng, and Y. Ma, "Runtime support for type-safe and context-based behavior adaptation," *Frontiers of Computer Science*, vol. 8, no. 1, pp. 17–32, 2014.

[8] M. Madsen, R. Zarifi, and O. Lhoták, "Tail call elimination and data representation for functional languages on the java virtual machine," in *Proceedings of the 27th International Conference on Compiler Construction—CC 2018*, pp. 139–150, ACM, New York, NY, USA, 2018.

[9] S. Papadimitriou, *Scientific Computing with ScalaLab at the Java Platform*, Scholars Press, Riga, Latvia, 2017.

[10] T. Cohen and J. Y. Gil, "Three approaches to object evolution," in *Proceedings of the 7th International Conference on Principles and Practice of Programming in Java–PPPJ'09*, pp. 57–66, ACM, New York, NY, USA, 2009.

[11] S. Malabarba, R. Pandey, J. Gragg, E. Barr, and J. F. Barnes, "Runtime support for type-safe dynamic java classes," in *Proceedings of the European Conference on Object-Oriented Programming—ECOOP'00*, pp. 337–361, Sophia Antipolis and Cannes, France, June 2000.

[12] Y.-L. Zhao, C.-P. Zhu, B. Han, and Q.-H. Zeng, "A calculus using fitness testing for method redirection," *Journal of Software*, vol. 24, no. 7, pp. 1495–1511, 2014.

[13] J. J. P. C. Rodrigues, D. B. De Rezende Segundo, H. A. Junqueira et al., "Enabling technologies for the internet of health things," *IEEE Access*, vol. 6, pp. 13129–13141, 2018.

[14] K. Ullah, M. A. Shah, and S. Zhang, "Effective ways to use internet of things in the field of medical and smart health care," in *Proceedings of the 2016 International Conference on Intelligent Systems Engineering (ICISE)*, pp. 372–379, Islamabad, Pakistan, January 2016.

[15] J. L. Kernec, F. Fioranelli, S. Yang, J. Lorandel, and O. Romain, "Radar for assisted living in the context of internet of things for health and beyond," in *Proceedings of the IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 163–167, Verona, Italy, October 2018.

[16] L. Haoyu, L. Jianxing, N. Arunkumar, A. F. Hussein, and M. M. Jaber, "An IoMT cloud-based real time sleep apnea detection scheme by using the SpO2 estimation supported by heart rate variability," *Future Generation Computer Systems*, vol. 98, pp. 69–77, 2019.

[17] L. Bettini, V. Bono, and B. Venneri, "Delegation by object composition," *Science of Computer Programming*, vol. 76, no. 11, pp. 992–1014, 2011.

[18] L. Baresi, C. Ghezzi, X. Ma, and V. P. L. Manna, "Efficient dynamic updates of distributed components through version consistency," *IEEE Transactions on Software Engineering*, vol. 43, no. 4, pp. 340–358, 2017.

[19] Y. Zhao, "Granule-oriented programming," *ACM SIGPLAN Notices*, vol. 39, no. 12, pp. 107–118, 2004.

[20] C. Quinton, R. Rabiser, M. Vierhauser, P. Grünbacher, and L. Baresi, "Evolution in dynamic software product lines: challenges and perspectives," in *Proceedings of the 19th International Conference on Software Product Line–SPLC'15*, pp. 126–130, ACM, New York, NY, USA, 2015.

[21] R. Hirschfeld, A. Igarashi, and H. Masuhara, "ContextFJ: A minimal core calculus for context-oriented programming," in *Proceedings of the Foundations of Aspect-Oriented Languages FOAL*, pp. 19–23, Porto de Galinhas, Brazil, March 2011.

Hindawi
Security and Communication Networks
Volume 2019, Article ID 6803801, 14 pages
https://doi.org/10.1155/2019/6803801

Research Article

Dynamics on Hybrid Complex Network: Botnet Modeling and Analysis of Medical IoT

Mingyong Yin,[1,2] **Xingshu Chen**,[3,4] **Qixu Wang**,[3,4] **Wei Wang**,[4] **and Yulong Wang**[2,3]

[1]*College of Computer Science and Technology, Sichuan University, Chengdu 610065, China*
[2]*Institute of Computer Application, Mianyang 621900, China*
[3]*College of Cybersecurity, Sichuan University, Chengdu 610065, China*
[4]*Cybersecurity Research Institute, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Qixu Wang; qixuwang@scu.edu.cn

Received 22 May 2019; Accepted 28 July 2019; Published 18 August 2019

Guest Editor: Kuan Zhang

With the rapid development of Internet of things technology, the application of intelligent devices in the medical industry has become ubiquitous. Connected devices have revolutionized clinicians and patient care but also made modern hospitals vulnerable to cyber attacks. Among the security risks, botnets are of particular concern, which can be used to control thousands of devices for remote data theft and equipment destruction. In this paper, we propose a non-Markovian spread dynamics model to understand the effects of botnet propagation, which can characterize the hybrid contagion situation in reality. Based on the Susceptible-Adopted-Recovered model, we introduce nonredundant memory spread mechanism for global propagation, as a tuner to adjust spreading rate difference. For describing the proposed model, we extend a heterogeneous edge-based compartmental theory. Through extensive numerical simulations, we reveal that the growth pattern of the final adoption size versus the information transmission probability is discontinuous and how the final adoption size is affected by hybrid ratio α, global scope control factor ϵ, accumulated received information threshold T, and other parameters on ER network. Furthermore, we give the theory and simulation result on BA network and also compare the two hybrid methods—single infection in one time slice and double infections in one time slice—to evaluate the influence on final adoption size. We found in SIOT hybrid contagion scenario the final adoption size shows the phenomenon of a decline followed by an increase versus different hybrid ratio, and it is both verified in theory and numerical simulation. Through validation by thousands of experiments, our developed theory agrees well with the numerical simulations.

1. Introduction

With the wide application of Internet of things (IoT) devices in the medical industry, security threats caused by limited computing power of devices, less security protection measures, and insufficient attention are also increasing, among which botnet is one of the biggest security threats. As is well known, most medical equipments have following security characters: always online, weak security protection, low cost of botnet attacks, and difficulty in clarifying the attribution of security responsibilities. With the control of medical IoT devices, botnet can be used to steal information and destroy devices according to hacker instructions. The security weaknesses of medical device can be manipulated to appropriate control over personal devices, hospital diagnostic machines, and other medical appliances. The work conducted by Jay Radcliffe in 2011 on weaknesses found in insulin pumps had aroused a lot of attention, he gave a live demonstration showing that it was possible to remotely deliver lethal doses of insulin to patients [1, 2].

Botnet, as a general bearing platform, has become the source of all kinds of network attacks. Botnet is evolved from traditional malicious code, which combines various attack methods, and has gradually become a highly efficient attack platform. Through botnet, the attacker implants botnet programs into the host in the network, controls the infected host, and establishes command and control channels. The biggest difference between botnet and traditional attacks is the one-to-many control structure, which enables attackers to control a large number of resources to serve them at a very low cost, which poses a huge

challenge to the security, confidentiality, and integrity of the medical industry network environment.

With the increasing threats of botnet, from antivirus companies to research institutions have conducted a large number of in-depth analysis and research on botnet, including botnet detection, tracking, defense, and countermeasures, and also, different defending mechanisms are introduced into IoT network [3–6]. The establishment of botnet propagation model is an effective tool to analyze and study the propagation characteristics of botnet, which is a necessary condition to understand the dynamics of the threat they pose. The recent frequent extortion of ransomware, such as Wannacry, Petya, etc., has caused great losses to individuals and enterprises. This kind of virus based on botnet can spread both on WAN and on LAN, and their propagation law also presents some new characteristics. It is a challenging problem to evaluate the influence of different information transmission channels on user adoption, the possibility of virus email sent by friends or strangers to be clicked and opened.

A mixture of local propagation and global propagation is typical in hybrid propagation mechanism, as depicted in Figure 1. For local propagation, where the infected node only infects a subset of the limited propagation target nodes, the infected node typically infects neighboring nodes [7]; for global propagation, the nodes are fully mixed, and the infected nodes can infect any other node [8, 9]. In fact, many epidemics use mixed transmission, which involves two or more combined transmission mechanisms. Also, the ransomware can scan a target computer on a local network or any computer randomly selected on the internet through a port scan. Among them, the local area network node is in the internal network environment which means the communication between internal nodes will not pass through firewall; also, to the node homogeneity, the probability of successful infection is higher in local spreading. Because the WAN node is not aware of the network environment of the target node, its success probability will be lower than the local success probability.

Another phenomenon we are interested in is when a host receives a number of disguised emails with viruses, the probability of computer infection will also increase because it is more likely to be misclicked; this memory effect makes the dynamics of social contagion non-Markovian.

In a word, in order to effectively depict the dissemination of botnet, we need to be able to describe the heterogeneous credibility of information from different sources, the impact range of different masters in the dissemination model, and the mixture way of different propagation method in single time slice. This characteristic also exists widely in other information and behavior dissemination. Therefore, it is necessary to study this dissemination scenario in order to provide a theoretical basis for the prediction and control of bot dissemination.

This paper proposes a hybrid propagation dynamics theoretical model based on the SAR model and the edge compartmental theory that includes local and global propagation and can capture differences in its propagation capabilities, which contributes in the following areas:

(1) In order to describe the phenomenon of botnet mixed propagation through LAN and WAN, we propose a hybrid propagation model that supports the global spreading participation node range control, which can better reflect the reality that a target node is infected by a limited range of attack nodes because of the impact of time, space, and randomness. It is different from previous work regarding global spreading as the infected node will infect every node in the network.

(2) We introduce nonredundant memory features in global propagation process, by setting the parameter of cumulative information that needs to be received for triggering state change; the propagation rate can be modified flexibly.

(3) Theoretical analysis and simulation experiments verify the effects of different mixing ratios on the final propagation range and find that under a certain spreading rate, the final propagation range will present a wavy curve phenomenon versus hybrid ratio α.

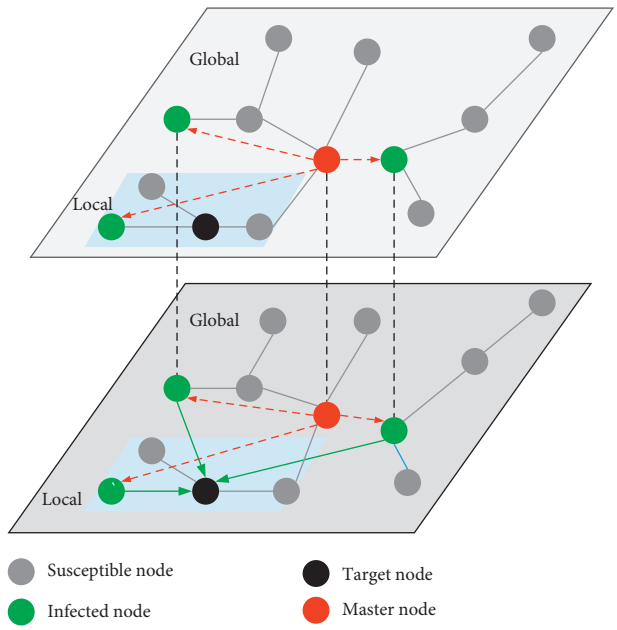


Figure 1: Illustration of hybrid propagation.

This paper is organized as follows. Section 2 gives a brief summary about related work on botnet propagation model. In Section 3, we abstract the scenes of different types of mixed propagation and present the model description. Based on the definition in Section 3, we give the theoretical derivation in Section 4. In Section 5, the correctness of the theory is verified by numerical simulation and program simulation, and the influence of different parameters on the final propagation range in the mixed propagation process is analyzed.

2. Related Work

2.1. Botnet Propagation Model. With the widespread use of IoT technology in the medical industry, ubiquitous smart

devices have greatly increased the attack surface while providing convenience for doctors and patients [10]. Among them, the malware infects the sensor and the terminal and is commanded and controlled by the external botnet master node, so the attacker initiates the attack to achieve purpose when the attacker needs it. These botnets composed of botnet devices have become the main threat to the network security and life safety in the medical industry, and they can breakthrough defense under heterogeneous network structure and different layers [11–14]. We need to perceive and understand the propagation process as early as possible to provide theoretical support for better control in different scenarios.

Botnet can generally be divided into infection, command and control, and attack phases. This article focuses on the infection phase of botnet. At this stage, attackers can spread bots in various ways, such as trojans, malicious emails, active scanning, passively inducing users to download and install bots, or proactively exploiting remote service vulnerabilities. After the attacker infects the target host, the hidden module is loaded, and the botnet program is hidden in the controlled host by techniques such as deformation and polymorphism. One of the most influential botnets is the Mirai botnet. Mirai uses worm-based propagation, which includes Internet of things cameras, routers, printers, and video recorders [1].

Modeling the botnet propagation based on the biological disease propagation model is a common method adopted by researchers. The propagation dynamics is used to model the propagation behavior and derive botnet spreading differential equations and then verify the worm propagation law with numerical simulation. They also try to solve the problem of how the network defender can prevent the formation of botnet by enhancing the security defense capability of the device under the condition that the network operation overhead is minimal [15–17].

Researchers have conducted extensive research on the propagation behavior of worms in wireless sensor networks. Representative work includes the Susceptible-Exposure-Infection-Recovery-Sensitivity Vaccination (SEIRS-V) model and the Susceptible-Exposure-Infection-Recovery-Vaccination (SEIRV) model. By capturing the spatiotemporal dynamics of the worm's propagation process, these models define equilibrium points using the basic reproductive number $R0$ and then assess the stability of the system at these points [18, 19].

Dagon et al. [20] discovered the law of botnet propagation affected by time and region based on the continuous monitoring of botnet and constructed a diurnal propagation model to characterize botnet infection. Todd Gardner et al. [21] researched botnet from the perspective of user behavior and found that we can mitigate the frequency of IoT botnet attacks with improved user information, which may positively affect user behavior; this can be used to predict user behavior after the botnet attack.

2.2. Dynamics on Complex Network. With the development of complex networks and communication dynamics, many phenomena in the fields of computer science, biology, sociology, and economics are characterized by "propagation dynamics on complex networks," and the methods to reveal their propagation laws are widely used [22–24].

In the field of Internet, the recent frequent extortion of ransomware, such as Wannacry, Petya, Scarab, etc., has caused great losses to individuals and enterprises [25–27]. For rumor spreading, ordinary users often receive opinions from opinion leaders and people they are familiar with; it is a challenging problem to evaluate the influence of different information spreading ways on user adoption. Therefore, it is necessary to conduct research on this hybrid propagation phenomenon and understand its law of transmission so as to further take effective countermeasures.

Research on social contagion mechanism and corresponding control strategies is one of the hotspots of current research. At present, scholars have carried out a lot of research on the impact of the heterogeneity of individual adoption behavior, heterogeneity of network structure, memory of individual adoption behavior, nonredundant contagion, and incomplete neighborhood spreading on social propagation. In reality, memory usually plays an important role on adoption enhancement for social contagion. For instance, when someone hears a message from many people, it is believed that the credibility of information will be greatly improved. When receiving a number of disguised emails with viruses, the probability of computer infection will also increase because it is more likely to be misclicked. This memory effect makes the dynamics of social contagion non-Markovian. Considering the memory effect, a modeling method based on non-Markov model is proposed in [28–30]. Generally speaking, a node can receive the cumulative information about specific social behavior either in a redundant or nonredundant manner [31], where the former allows a pair of individuals to successfully transmit information many times, but for the latter, repetitive transmission is prohibited. Previous studies on nonredundant information transmission characteristics of society have been relatively few [30, 32, 33]. It is of great significance to understand the dynamics of transmission with nonredundant information memory effect in hybrid spreading situation.

3. Model Descriptions

In this section, we give the model of botnet propagation in hybrid spreading scenario, to characterize the comprehensive effect on target node. It can reflect the fact that medical devices can be infected by local area nodes or internet terminals with different impacts. For the network G composed of N nodes, the average degree is $\langle k \rangle$ and the degree of node i is k_i. The nodes participate in one of the propagations with a certain probability in each time slice. For node i, during each round of propagation, it will involve in local propagation or global propagation with probability α; we note this kind of propagation as single in one time slice (SIOT). Correspondingly, for the situation that node i can receive messages from both local and global propagation, we name it as double in one time slice (DIOT).

In order to describe the heterogeneous credibility of information received from the local and the global sources in the case of mixed propagation, we assume that the local

propagation threshold and the global propagation threshold are different and the global propagation information is received as a nonredundant memory process; each node information can only be passed once to the target node. As shown in Figure 1, the target node (black) can receive information from the local neighbor infected nodes (green) and global infected nodes.

In the case of local propagation, nodes are more likely to adopt corresponding ideas or infect similar viruses, so we set the threshold of local contagion to 1. The infected neighbor node j infects node i with probability lambda; that is, the probability of accepting information from local propagation per round is $\lambda_L = \alpha\lambda$. Similarly, node i participates in global propagation with probability $1-\alpha$, and the rate is $\lambda_G = (1-\alpha)\lambda$. Due to the large number of nodes in the whole network, we introduce the global parameter ϵ to control the node scale in propagation; the global node participating in the propagation of i is $N\epsilon$, and the number of participating global nodes can be adjusted by the parameter ϵ. In the global propagation situation, the Internet attack node randomly scans the target user for botnet propagation and randomly sends the message for propagation.

Compared with local propagation, the information credibility from global channel is less trustworthy. Therefore, we set the threshold as T, and it is satisfied that each node receives broadcast information of other nodes no more than once. In addition, since the number of global nodes is much larger than that of local nodes, in the modeling process, to simplify processing, the global propagation node includes neighbor nodes of node i.

For the dynamic modeling of network propagation process, this paper references the SAR (Susceptible-Adopted-Recovered) model. At any time, any node in the network is in one of these three states, as shown in Figure 2. S represents susceptible state, indicating that a node in the network can be infected; A represents infected state, indicating that a node in the network has been infected; and R represents recovery state, indicating that the infected node in the network has changed to a recovery state and can no longer participate in the follow-up process.

In each propagation round, we assume that one node can participate in either global or local contagion one time if the node state is S; for nodes in A state, it can try once for recovering to R state by sampling γ. For the mixed propagation of different intensity propagation sources, we are concerned about the outbreak threshold characteristics, especially the first-order phase transition. We further investigate the impact on the final adoption size under different hybrid ratios of mixed propagation, various transmission rates, and initial seed ratio during the propagation process.

4. Theory

4.1. SIOT. In this section, we make use of generalized heterogeneous edge-based compartmental theory, based on the previous work in [34–36] to describe our model and characterize the hybrid propagation process based on edge-based compartmental theory for the analysis. Although the system in [35] was proposed to analyze single-mechanism-based spreading for the continuous time case, it can be modified to be suitable for our model with hybrid propagation for discrete time and nonredundant information memory characteristic. We calculate the probability that a random test node u is in each state: susceptible $S(t)$, infected $A(t)$, and recovered $R(t)$.

We define the probability that a node has degree k is $p(k)$; it means the number of neighbors of node u for local spreading is k. The generating function of degree distribution $p(k)$ is defined as $g(x) = \sum_k p(k)x^k$, where $p_n(k)$ means the probability that, for a random neighbor of u, it has k edges. We assume the degrees of the two end nodes of each edge are independent.

In an uncorrelated network $p_n(k) = kp(k)/\langle k\rangle$, where $\langle k\rangle$ is the average degree of the network, we denote θ_t as the probability that a random neighbor v has not infected u through local path. Let ϑ_t be the probability that global node w has not infected u through global path.

Suppose u has k neighbors, the probability that it is susceptible is decided by local and global spreading result. For local propagation, we assume the infection threshold is 1, i.e., whenever node u receives one message from neighbors, it will be infected, so we can get $S_L(\overrightarrow{k}, t) = \theta_t^k$ for nodes which have degree k. For global propagation, influenced by the factors like low trust and environment heterogeneity, we assume the infection threshold is T, and T is greater than or equal to 1; at time t, the probability of node u not infected through global spreading is

$$S_G(\overrightarrow{k}, t) = \binom{N-1}{m}\vartheta_t^{N-1-m}(1-\vartheta_t)^m, \tag{1}$$

where n is the number of nodes attending in the propagation. So, at time t, the probability that node u is in the susceptible state can be written as

$$S(\overrightarrow{k}, t) = \theta_t^k \sum_{m=0}^{T-1}\binom{N-1}{m}\vartheta_t^{N-1-m}(1-\vartheta_t)^m. \tag{2}$$

Then, by averaging $S(\overrightarrow{k}, t)$ over all degrees, the initial ratio of nodes in adopted state is ρ_0, and we have

$$S(t) = (1-\rho_0)\sum_k p(k)\theta_t^k \sum_{m=0}^{T-1}\binom{N-1}{m}\vartheta_t^{N-1-m}(1-\vartheta_t)^m. \tag{3}$$

A neighbor of individual u may be in one of susceptible, adopted, or recovered states. We can thus further express θ_t as

$$\theta_t = \phi_S(t) + \phi_A(t) + \phi_R(t), \tag{4}$$

where $\phi_S(t), \phi_A(t), \phi_R(t)$ is the probability that a neighbor of the individual u, is in the state of susceptible, adopted, or recovered, and has not transmitted the information to

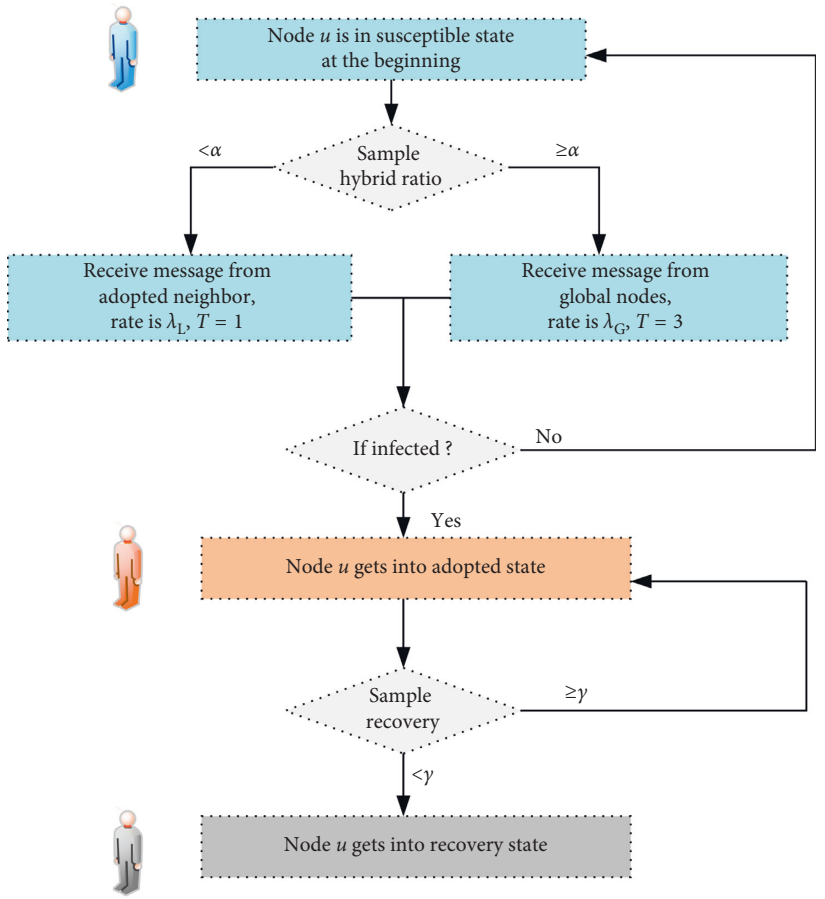


FIGURE 2: The flow chart of node state transferring; in each spread phase, a node will act in either local or global propagation according to the sample result.

individual u by time t. We need to seek the solution of three possibilities. Assume a neighboring individual v of u is in the susceptible state at start point; it cannot transmit the information to u. Individual v can get the information from its other neighbors, since u is in a cavity state. Neighbor v cannot be infected by u and itself; then,

$$\phi_S(t) = (1-\rho_0)\sum_k kp(k)\theta_t^{k-1}\binom{N-2}{m}\sum_{m=0}^{T-1}\vartheta_t^{N-2-m}\frac{(1-\vartheta_t)^m}{\langle k\rangle}. \tag{5}$$

We further investigate $\phi_R(t)$; it should satisfy the definition that an adopted neighbor has not transmitted the information to u via its connection and with probability γ the adopted neighbor to be recovered. According to the analysis above, we get

$$\frac{d\phi_R(t)}{dt} = \gamma(1-\lambda_L)\phi_A(t). \tag{6}$$

At time t, the rate of change in the probability that a random edge has not transmitted the information is equal to the rate at which the adopted neighbors transmit the information to their susceptible neighboring individuals through edges. Thus, we get

$$\frac{d\theta(t)}{dt} = -\lambda_L\phi_A(t). \tag{7}$$

Combining equations (6) and (7), we obtain

$$\phi_R(t) = \frac{\gamma(1-\theta(t))(1-\lambda_L)}{\lambda_L}. \tag{8}$$

$$\frac{d\theta(t)}{dt} = -\lambda_L(\theta(t)-\phi_S(t)-\phi_R(t)). \tag{9}$$

Substitute equations (8) and (5) into equation (7). Doing so, we can rewrite equation (6) as

$$\frac{d\theta(t)}{dt} = \lambda_L\sum_k\frac{kp(k)}{\langle k\rangle}\theta_t^{k-1}\binom{N-2}{m}\sum_{m=0}^{T-1}\vartheta_t^{N-2-m}(1-\vartheta_t)^m + \gamma(1-\theta(t))(1-\lambda_L) - \lambda_L\theta(t). \tag{10}$$

We can write ϑ_t as

$$\vartheta_t = \varphi_S(t) + \varphi_A(t) + \varphi_R(t). \tag{11}$$

In the same way with local spreading, for global propagation, we take into account the weak relationship with global nodes; the threshold for state change from

susceptible state to adopted state is T, that is, a node should at least receive T messages from global spreading and then it can trigger state change. Then, $\varphi_S(t)$ can be written as

$$\varphi_S(t) = (1-\rho_0)\sum_K p(k)\theta_t^k \binom{N-2}{m} \sum_{m=0}^{T-1} \vartheta_t^{N-2-m}(1-\vartheta_t)^m. \tag{12}$$

So $\varphi_R(t)$ is

$$\varphi_R(t) = \frac{\gamma(1-\vartheta(t))(1-\lambda_G)}{\lambda_G}. \tag{13}$$

Then, we can get

$$\frac{d\vartheta(t)}{dt} = \lambda_G \sum_K p(k)\theta_t^k \binom{N-2}{m} \sum_{m=0}^{T-1} \vartheta_t^{N-2-m}(1-\vartheta_t)^m + \gamma(1-\vartheta(t))(1-\lambda_G) - \lambda_G\vartheta(t). \tag{14}$$

We know $S(t) + A(t) + R(t) = 1$ at time t, note that the rate $dA(t)/dt$ is equal to the rate at which $S(t)$ decreases because all the individuals moving out of the susceptible state must move into the adopted state minus the rate at which adopted individuals become recovered. We have

$$\frac{dA(t)}{dt} = -\frac{dS(t)}{dt} - \gamma A(t), \tag{15}$$

$$\frac{dR(t)}{dt} = \gamma A(t). \tag{16}$$

According to the deduction above, we can have the general description of social contagion dynamics so that we can calculate the probability that node u has not received enough messages for state changing.

$$\theta(\infty) = \sum_k \frac{kp(k)}{\langle k\rangle}\theta_{(\infty)}^{k-1} \binom{N-2}{m} \sum_{m=0}^{T-1} \vartheta_{(\infty)}^{N-2-m}(1-\vartheta_\infty)^m + \frac{\gamma(1-\theta(\infty))(1-\lambda_L)}{\lambda_L}, \tag{17}$$

$$\vartheta(\infty) = \sum_k p(k)\theta_\infty^k \binom{N-2}{m} \sum_{m=0}^{T-1} \vartheta_\infty^{N-2-m}(1-\vartheta_\infty)^m + \frac{\gamma(1-\vartheta(\infty))(1-\lambda_G)}{\lambda_G}. \tag{18}$$

Now, we analyze the critical information transmission probability. Since we have already assumed that $T > 1$ to study the memory reinforcement, a vanishingly small fraction of seeds cannot trigger a global behavior adoption. In this situation, $\theta_x(\infty) = 1$ is not a solution of the following equation:

$$\frac{\partial f_L(\theta(\infty),\vartheta(\infty))}{\partial\theta(\infty)} \frac{\partial f_G(\theta(\infty),\vartheta(\infty))}{\partial\vartheta(\infty)} = 1. \tag{19}$$

From theory analysis, we can capture first-order phase transition at the critical point, where the condition is fulfilled. We assume $A(\infty) = 0$, then $R(\infty) = 1 - S(\infty)$; we can calculate $R(\infty)$ as final adoption size.

4.2. DIOT. For the theory introduced above, it assumes that a node can participate in only one type of spreading in each time slice, either local or global propagation. In reality, different propagation may act on nodes at the same time, so we also carry out research on this scenario.

In alternative hybrid contagion, the spreading rate for local and global propagation is $\lambda_L = \alpha\lambda$ and $\lambda_G = (1-\alpha)\lambda$, respectively, while $\lambda_L + \lambda_G = \lambda$. Different from alternative hybrid contagion, the spreading rate of parallel hybrid contagion does not have such constraints, and λ_L and λ_G are isolated.

To further explore the contribution of two spreading methods in hybrid propagation, we introduce globe spreading rate control factor ζ; let $\lambda_G = \lambda_L/\zeta$; by doing this, we can get the variety of final adoption size versus different global transmission rate. So, equations (17) and (18) can be written as

$$\theta(\infty) = \sum_k \frac{kp(k)}{\langle k\rangle}\theta_{(\infty)}^{k-1} \binom{N-2}{m} \sum_{m=0}^{T-1} \vartheta_{(\infty)}^{N-2-m}(1-\vartheta_\infty)^m + \frac{\gamma(1-\theta(\infty))(1-\lambda)}{\lambda}, \tag{20}$$

$$\vartheta(\infty) = \sum_k p(k)\theta_\infty^k \binom{N-2}{m} \sum_{m=0}^{T-1} \vartheta_\infty^{N-2-m}(1-\vartheta_\infty)^m + \frac{\gamma(1-\vartheta(\infty))(1-\lambda/\zeta)}{\lambda/\zeta}. \tag{21}$$

5. Simulation

5.1. Simulation Method. Based on the theory analysis of the botnet spreading progress, we perform numerical simulations to study our proposed hybrid contagion model, using Erdos–Renyi (ER) network model [37] and Barabasi–Albert (BA) network with power-law degree distribution for our simulations [8]. For medical IoT, the medical equipment or sensors are always deployed in diagnosis and treatment room or datacenter; in general, it is hard to infect them by email attachments as commonly seen in computer. The most possible attack vector is wired or wireless network intrusion and hardware addition by human intervention, which can be categorized as local propagation; we can model these possible propagation channels with hybrid spreading model. An overview of the proposed numerical simulation program is shown in

Algorithm 1. In initiation phase, ER network generation and parameter settings need to be handled first. We use the open-source package NetworkX [38] to produce network ER network G, the network size is 10,000 network nodes, and the average degree is $\langle k \rangle = 10$.

We randomly set 5 nodes in adopted state, $\rho_0 = 5/10^4$. At each experiment, according to the variable that needs to be investigated, parameters like local propagation probability λ_L, global propagation probability λ_G, threshold T, recovery rate γ, hybrid ratio α, and globe scope controller ϵ are set respectively. In most cases, we set the scope parameter $\epsilon = 0.004$, that is, in each transmission, node u will receive messages from 40 global nodes. For each experiment, we repeat a thousand times and take the average value as simulation result.

5.2. SIOT in ER Network. We first study the effects of hybrid ratio α on social contagions in ER networks. As shown in Figure 3, the hybrid ratio changes the growth pattern of the final behavior adoption size $R(\infty)$ versus the information transmission probability λ. From the figure, we can see that when $\lambda = 0.1$, the final adoption size $R(\infty)$ is varied with α increments. When α value is small, the local propagation contributes less, and it is hard to outbreak when initial seeds are few. Nonetheless, when α gets higher, more chances are there for the node to receive message from neighbors; as we aforementioned, the threshold is 1, so it will promote the probability of nodes in susceptible state to get into adopted state. When more nodes are in adopted state, for global propagation, it is much easier to receive more messages than threshold for state changing. Furthermore, when α keeps on augmenting larger than the outbreak value, the final adoption size will gradually decline and ascend afterwards. Our theoretical predictions agree well with the numerical results. The differences between the theoretical and numerical predictions are caused by the strong dynamical correlations among the states of neighbors.

We further identify the outbreak threshold by the variability measure, which is a standard measure to determine the critical point in equilibrium phase on magnetic system, to reflect the fluctuation of the outbreak size for different α:

$$\delta = \frac{\sqrt{R^2 - \langle R \rangle^2}}{\langle R \rangle}. \tag{22}$$

When we fix the hybrid ratio α, the growth pattern of $R(\infty)$ versus transmission rate λ can be observed Figure 4.

We further investigate the relation between hybrid ratio α and final adoption size $R(\infty)$'s variation law; by calculating the relative change rate of $R(\infty)$, we can derive the variation pattern. It can be seen from Figure 5 that with the increase of α, the burst threshold decreases, indicating that local propagation still plays a dominant role in the mixed propagation process. The variability exhibits a peak over a wide range of λ. In our model, we introduce parameter ϵ to control the size of nodes joining in global propagation; the reason behind this is although node u can receive message from any node in the global

```
Initialization:
(1)  Network generation
(2)  Parameters initialization
     begin:
(1)  newState[]<- hisState[]
(2)  for: any node n_i in N
(3)    if node state is susceptiblesample propagation
       method with α;
(4)    if local:
(5)      get neighbor nodes list from G and node state
         from hisState[];
(6)      for: any node in neighbor[]
(7)        if node state is infected, then:
(8)          infect node n_i with λ_L;
(9)          if count > 1, update newState[];
(10)   else if global:
(11)     get Nϵ global nodes global[] from G and
         node state from hisState[];
(12)     for: any node in global[]
(13)       if node state is infected:
(14)         infect node n_i with λ_G and update count of
             received messages;
(15)         if count > T, update newState[];
(16)   else if node state is infected:
(17)     to recover with probability γ;
(18)     update newState[];
(19)   hisState[]<- newState[];
       calculate R, and refresh loop parameters.
(20) end
     Output: final adoption size R
```

Algorithm 1: Numerical simulation pseudocode.

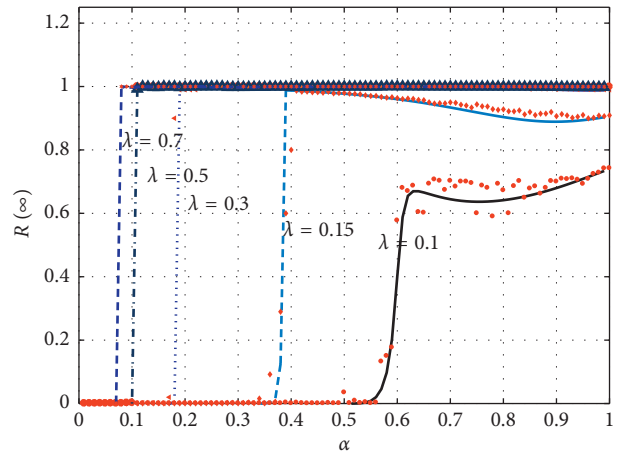


Figure 3: Propagation with fixed λ. The final behavior adoption size $R(\infty)$ versus the hybrid ratio α with fixed information transmission probability, $\lambda = 0.1, 0.15, 0.3, 0.5, 0.7$, respectively. The lines are the theoretical predictions and the dots are the simulation results.

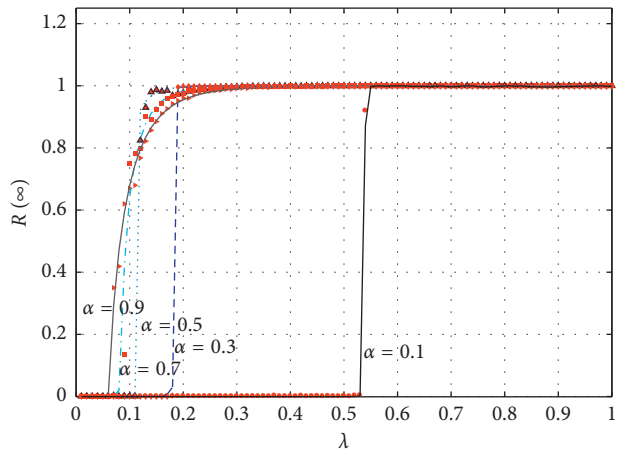


FIGURE 4: The final behavior adoption size $R(\infty)$ versus the information transmission probability λ, with fixed hybrid ratio, $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$, respectively. The lines are the theoretical predictions and the dots are the simulation results.

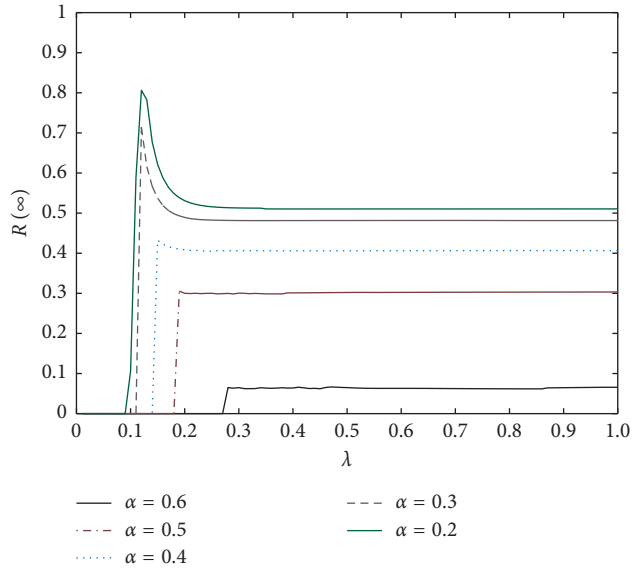


FIGURE 5: The variation of final adoption size $R(\infty)$ versus different λ with different hybrid ratio α.

method, in each round, it can be affected by only a few of them.

As shown in Figure 6, as ϵ increases, it is faster to reach the outbreak threshold value. In the same way, we further study the effects of γ on the spreading behavior. By setting $\alpha = 0.5$ and $\epsilon = 0.004$, we can investigate how the recovery rate γ influences final adoption size R, as shown in Figure 7. It visually demonstrated the change of outbreak threshold of λ; larger γ means slower outbreaks. Finally, we focus on the impact of the different memory threshold T on the propagation range. In our model, we use parameter T to adjust the information credibility, which means for large T value, more information needs to be received to change its status, as shown in Figure 8.

5.3. SIOT in BA Network. The BA network is one of the classical scale-free networks whose degree distribution follows a power law. The first scale-free model, the BA model, has a linear preferential attachment $\prod(k_i) = k_i/\sum_j k_j$ and adds one new node at every time step. Thus, in general, $\prod(k)$ has the form $\prod(k) = A + k^{\alpha}$, where A is the initial attractiveness of the node. We also set the network scale as 10,000 nodes, $\langle k \rangle = 10$, and $\rho = 5/10^4$. Other parameters are set as follows: threshold $T = 3$ for global contagion, recovery rate is $\gamma = 0.5$, and globe scope controller is $\epsilon = 0.004$.

Firstly, we can find in Figure 9 that nodes propagate faster in the BA network than in the ER network in the case of same average degree. Because BA network has unbiased degree distribution, large degree nodes have more neighbors to foster information propagation. We also find the phenomenon that final adoption size changes from decline to rise as α increases. Compared with the ER network, the BA network has 30% decrease when it reached the peak value; when $\lambda = 0.1$, greater amplitude of oscillation was caused by difference in degree

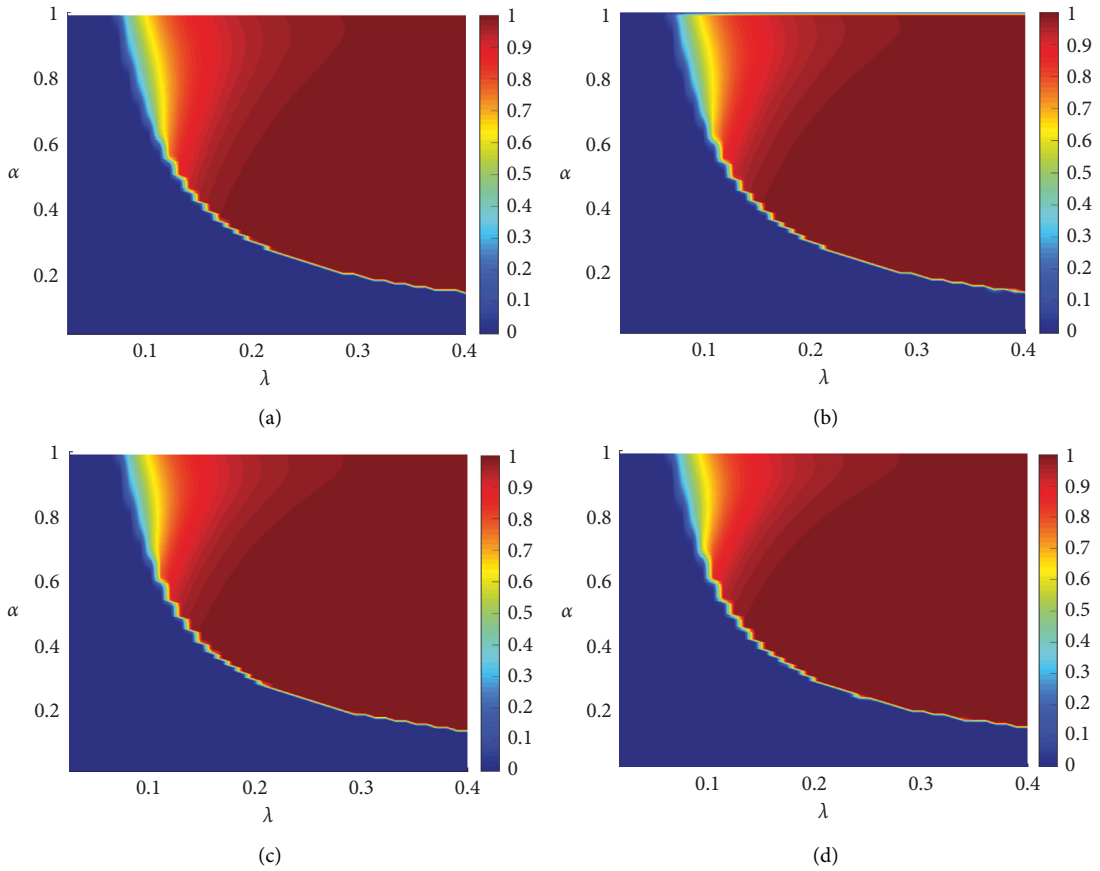


Figure 6: Final adoption size varied with ϵ. ϵ value is 0.0035, 0.004, 0.0045, and 0.005, respectively, from (a) to (d).

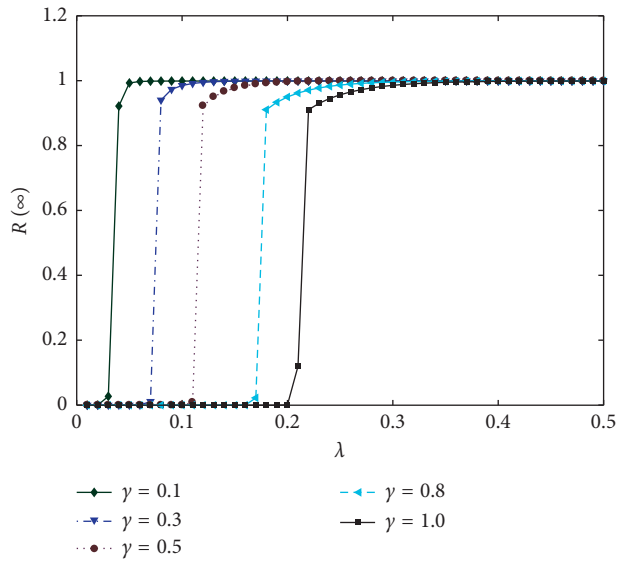


Figure 7: Final adoption size varied with γ while keeping other parameters unchanged.

distribution. Generally, the BA network can reach the burst threshold much faster than the ER network under the same lambda condition, as shown in Figure 10. In the same way, we fixed hybrid ratio α and observed the change of final adoption size with spreading rate; from Figure 11, it can be seen that when global propagation dominates, it spreads faster than the ER network, but when the local propagation ratio increases, the difference between these two network gets smaller.

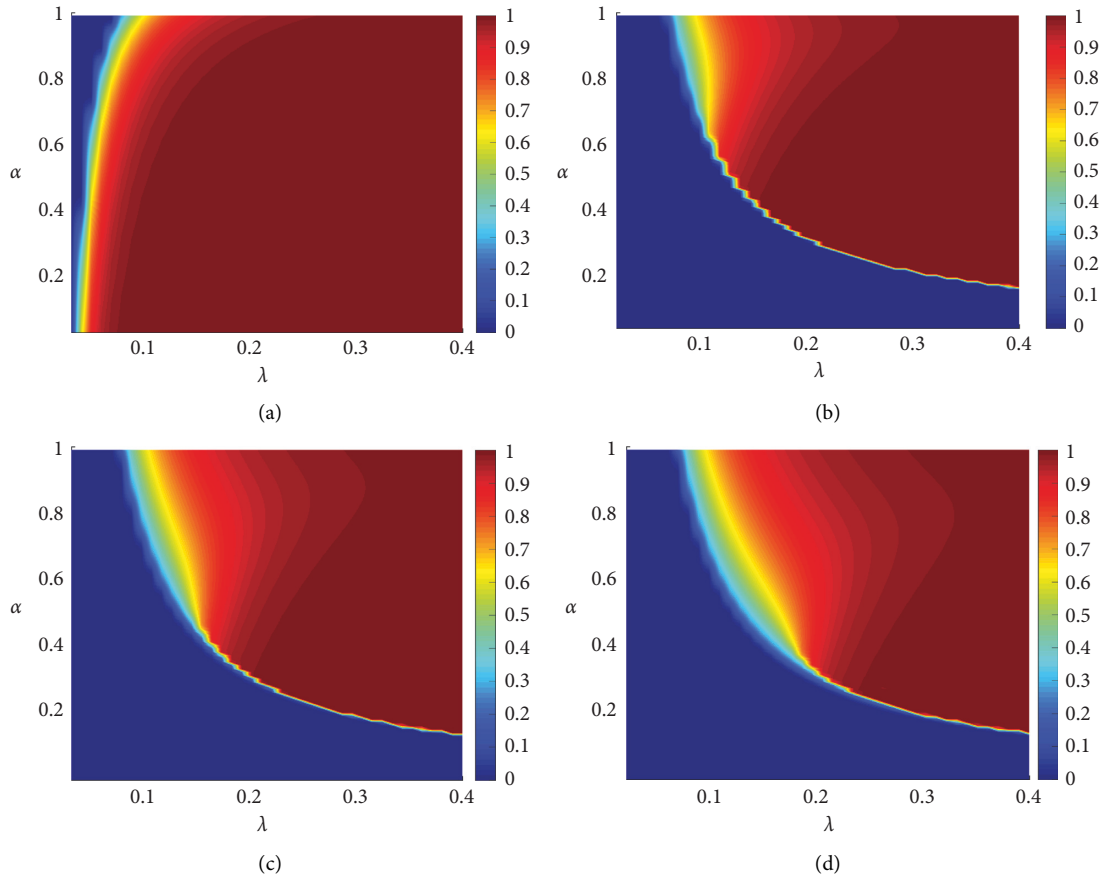


Figure 8: Final adoption size varied with T. (a) to (d) illustrate the result of $T=1$, $T=2$, $T=3$, and $T=4$, respectively.

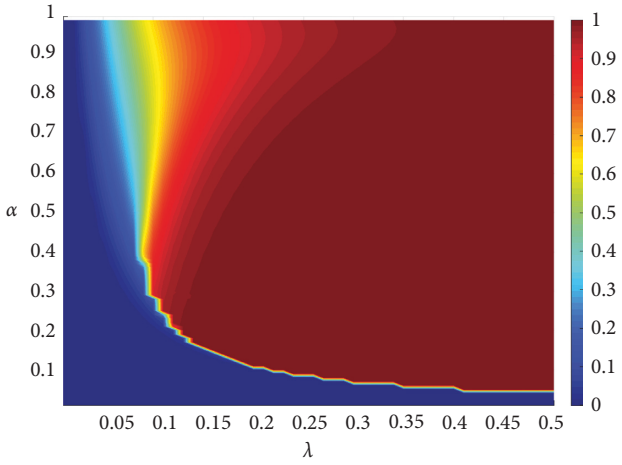


Figure 9: Final adoption size varied with ϵ in BA network. ϵ value is 0.0035, 0.004, 0.0045, and 0.005, respectively.

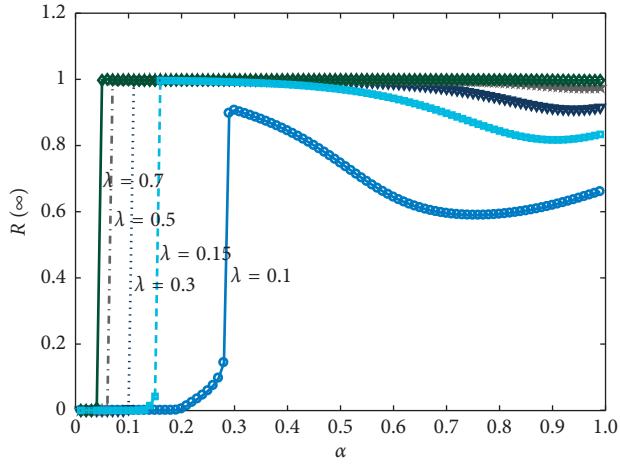


Figure 10: Final adoption size varied with α in SIOT.

5.4. DIOT. For the model introduced above, it assumes that a node can participate in only one type of spreading in each time slice, either local or global propagation. In reality, different propagations may act on nodes at the same time, so we also carry out research on this scenario.

In SIOT hybrid contagion, the spreading rate for local and global propagation is $\lambda_L = \alpha\lambda$ and $\lambda_G = (1-\alpha)\lambda$, respectively, while $\lambda_L + \lambda_G = \lambda$; the parameter α is used to adjust contagion attendance for different propagations. Compared with SIOT hybrid contagion, the spreading rate of DIOT does not have such constraints. λ_L and λ_G are isolated; this also means that a node can receive messages from local or global nodes in same time slice. The information transmission flow can be seen in Figure 12. Besides this trivial difference, other transmission parameters and contagion process are the same with SIOT hybrid

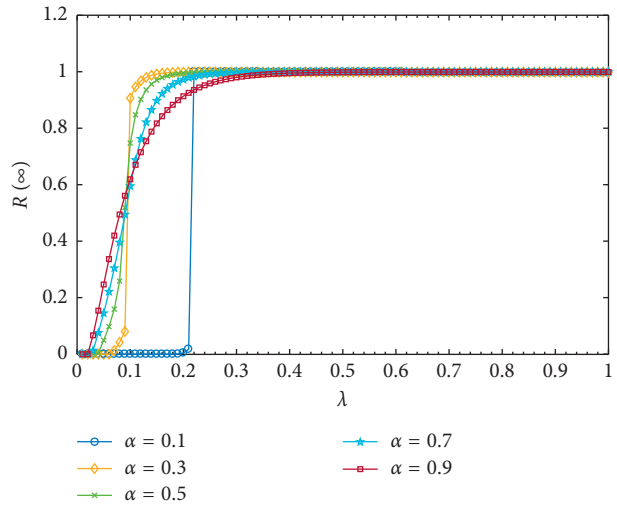


Figure 11: Final adoption size varied with λ in SIOT.

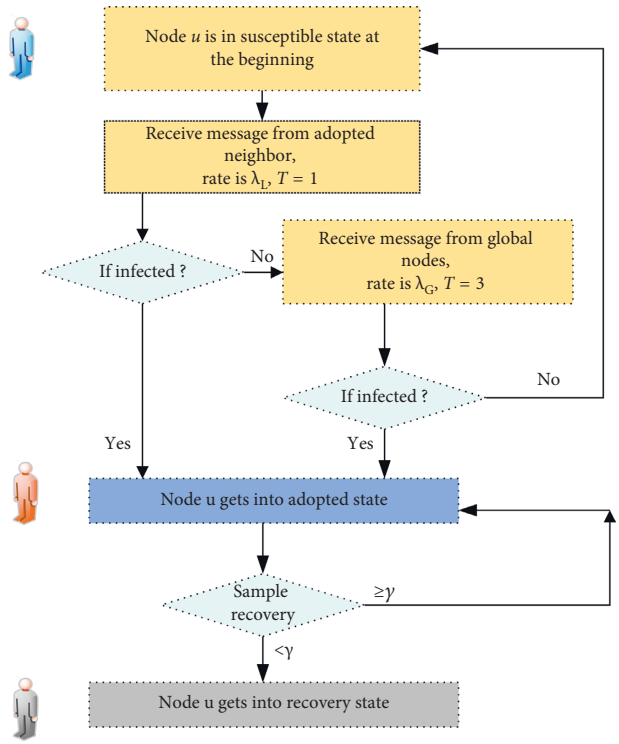


Figure 12: The flow chart of node state transferring; in each spread phase, a node will act in both local and global propagation.

transmission. For illustration convenience, we set $\lambda_L = \lambda$ and introduced global transmission rate scale parameter ζ to change transmission rate of λ_G, that is, $\lambda_G = \lambda/\zeta$; the value of ζ is from 1 to 100. As illustrated in Figure 13, we can find that nodes spread in the DIOT mode can reach outbreak threshold at lower λ than in the SIOT mode. When λ is larger than 0.12, it may reach the outbreak threshold, but if the value of λ is smaller than 0.005, it can never outbreak. The final adoption size $R(\infty)$ changes with ζ; as shown in Figure 14, we can find the multiple factor ζ can play a major role when its value is small, and as it increases, the global transmission rate will be too small to affect the adoption result. We can find from Figure 15 that changing the global spreading rate can vary the approach speed to full outbreak, but the critical point is the same, which means the discontinuous growth is controlled by local propagation.

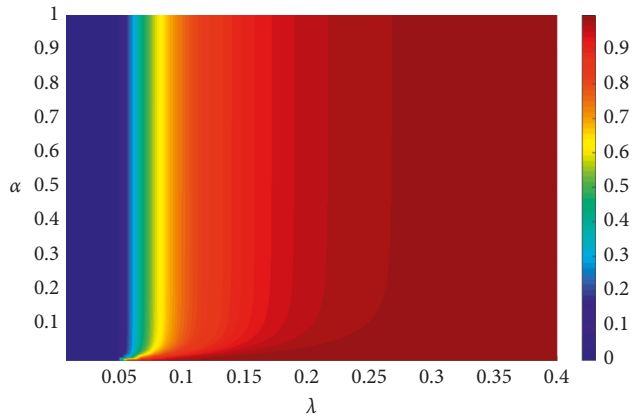


Figure 13: DIOT propagation. The final behavior adoption size $R(\infty)$ versus the global transmission rate scale and local transmission rate; other parameters are $N = 10{,}000$, $\epsilon = 0.004$, $T = 3$, and $\gamma = 0.5$, respectively. The lines are the theoretical predictions.

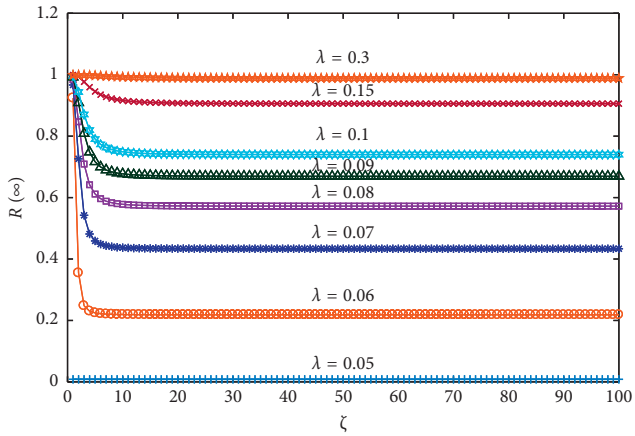


Figure 14: Final adoption size $R(\infty)$ varied with ζ in DIOT.

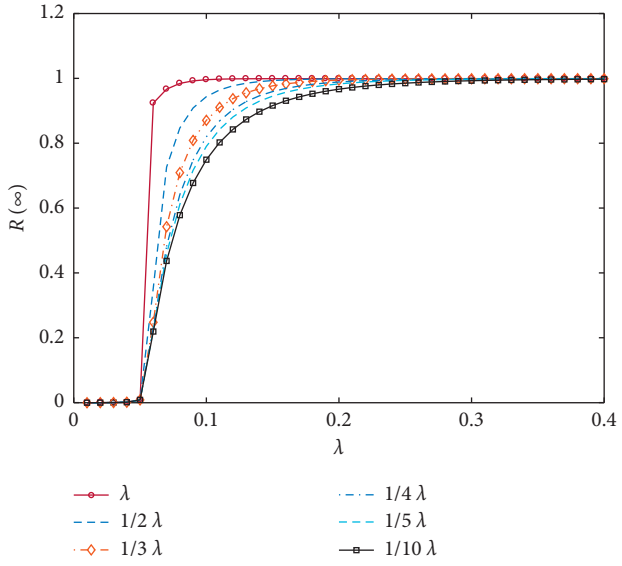


Figure 15: Final adoption size $R(\infty)$ varied with λ in DIOT.

6. Conclusion

In this paper, we studied the effects of hybrid propagation with different spreading rates and memory reinforcements on botnet contagions. We first proposed an information contagion model to describe the botnet spreading dynamics on complex networks. We then developed a generalized heterogeneous edge-based compartmental theory to describe the proposed model.

Through extensive numerical simulations on the ER network and BA network, we found that the growth pattern of the final behavior size $R(\infty)$ versus the hybrid ratio α exhibits discontinuous pattern when fixed transmission rate λ is large. But when λ is small, $R(\infty)$ shows the phenomenon of fluctuation, and at critical point, it reaches peak value first, followed with small amplitude declining and gradually rising. In addition, we also fixed the hybrid ratio α to analyze the final adoption size $R(\infty)$ changing with transmission rate λ, and the growth pattern of $R(\infty)$ changing from continuous to discontinuous is observed.

For comparing the effect of different hybrid methods, SIOT and DIOT are proposed, and the simulation result is presented; obviously, DIOT can spread faster especially when global transmission rate is high. We finally studied the effect of other parameters and found that memory threshold T, recovery rate γ, and global propagation range controller ϵ can affect $R(\infty)$ growing pattern, respectively; when T is small, it grows much faster because more seeds can be generated and global spreading can contribute more. With increasing γ , it gets slower to reach the burst value. Also, global range controller ϵ can change the pattern; when ϵ gets larger, it reaches critical value much faster. By introducing hybrid propagation mechanism and spreading scope controller, with memory character, the method can support modeling different spreading scenarios flexibly, but it simplifies the life states of bot, and the immune characteristics of nodes are not taken into account, so our future work will focus on these points.

Our proposed theory agrees well with the numerical simulations on ER and BA networks. The model proposed in this paper can provide theoretical reference for hybrid propagation modeling of botnet in complex networks and also provide guidance for medical industry to deal with botnet threats.

Data Availability

We conducted our experiment with the numerical simulation method, without using any open dataset.

Conflicts of Interest

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

Acknowledgments

This study was financially supported in part by a program of National Natural Science Foundation of China (NSFC)

(grant nos. 61272447 and 61802271) and in part by the Fundamental Research Funds for the Central Universities (grant nos. SCU2018D018 and SCU2018D022). This support is gratefully acknowledged.

References

[1] M. Antonakakis, T. April, M. Bailey et al., "Understanding the mirai botnet," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1093–1110, USENIX Security 17, Vancouver, BC, Canada, August 2017.

[2] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the IoT: mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[3] D. Chen, N. Zhang, R. Lu, N. Cheng, K. Zhang, and Z. Qin, "Channel precoding based message authentication in wireless networks: challenges and solutions," *IEEE Network*, vol. 33, no. 1, pp. 99–105, 2019.

[4] D. Chen, N. Zhang, Z. Qin et al., "S2M: a lightweight acoustic fingerprints-based wireless device authentication protocol," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 88–100, 2016.

[5] Q. Wang, D. Chen, N. Zhang, Z. Qin, and Z. Qin, "LACS: a lightweight label-based access control scheme in IoT-based 5G caching context," *IEEE Access*, vol. 5, pp. 4018–4027, 2017.

[6] K. Zhang, X. Liang, J. Ni, K. Yang, and X. S. Shen, "Exploiting social network to enhance human-to-human infection analysis without privacy leakage," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 607–620, 2016.

[7] C. Zhang, S. Zhou, J. C. Miller, I. J. Cox, and B. M. Chain, "Optimizing hybrid spreading in metapopulations," *Scientific Reports*, vol. 5, no. 1, p. 9924, 2015.

[8] R. M. Anderson, "Discussion: the Kermack-McKendrick epidemic threshold theorem," *Bulletin of Mathematical Biology*, vol. 53, no. 1-2, pp. 3–32, 1991.

[9] M. Newman, *Networks: An Introduction*, Oxford University Press, Oxford, UK, 2010.

[10] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pp. 268–273, IEEE, Athens, Greece, June 2009.

[11] A. Laha, N. Zhang, H. Wu, D. Chen, and T. Han, "Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5520–5530, 2019.

[12] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[13] D. Chen, N. Zhang, N. Cheng, K. Zhang, Z. Qin, and X. S. Shen, "Physical layer based message authentication with secure channel codes," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2018.

[14] Q. Wang, D. Chen, N. Zhang, Z. Ding, and Z. Qin, "PCP: a privacy-preserving content-based publish-subscribe scheme with differential privacy in fog computing," *IEEE Access*, vol. 5, pp. 17962–17974, 2017.

[15] D. Acarali, M. Rajarajan, N. Komninos, and B. B. Zarpelão, "Modelling the spread of botnet malware in IoT-based wireless sensor networks," *Security and Communication Networks*, vol. 2019, Article ID 3745619, 13 pages, 2019.

[16] M. Ajelli, R. Lo Cigno, and A. Montresor, "Modeling botnets and epidemic malware," in *Proceedings of the 2010 IEEE International Conference on Communications*, pp. 1–5, IEEE, Cape Town, South Africa, May 2010.

[17] M. J. Farooq and Q. Zhu, "Modeling, analysis, and mitigation of dynamic botnet formation in wireless IoT networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2412–2426, 2019.

[18] B. K. Mishra and D. K. Saini, "SEIRS epidemic model with delay for transmission of malicious objects in computer network," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1476–1482, 2007.

[19] A. Singh, A. K. Awasthi, K. Singh, and P. K. Srivastava, "Modeling and analysis of worm propagation in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 3, pp. 2535–2551, 2018.

[20] D. Dagon, C. C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proceedings of the NDSS Symposium 2006*, vol. 6, pp. 2–13, San Diego, CA, USA, February 2006.

[21] M. Todd Gardner, C. C. Beard, and M. Deep, "Using seirs epidemic models for IoT botnets attacks," in *Proceedings of the DRCN 2017—Design of Reliable Communication Networks*, pp. 1–8, Munich, Germany, March 2017.

[22] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Reviews of Modern Physics*, vol. 81, no. 2, pp. 591–646, 2009.

[23] J. C. Flack and R. M. D'Souza, "The digital age and the future of social network science and engineering," *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1873–1877, 2014.

[24] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of Modern Physics*, vol. 87, no. 3, pp. 925–979, 2015.

[25] V. Constantin Craciun, A. Mogage, and E. Simion, "Trends in design of ransomware viruses," in *International Conference on Security for Information Technology and Communications*, pp. 259–272, Springer, Berlin, Germany, 2018.

[26] 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.

[27] A. Zimba, L. Simukonda, and M. Chishimba, "Demystifying ransomware attacks: reverse engineering and dynamic malware analysis of wannacry for network and information security," *Zambia ICT Journal*, vol. 1, no. 1, pp. 35–40, 2017.

[28] S. Aral and D. Walker, "Identifying influential and susceptible members of social networks," *Science*, vol. 337, no. 6092, pp. 337–341, 2012.

[29] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson, "The diffusion of microfinance," *Science*, vol. 341, no. 6144, article 1236498, 2013.

[30] P. S. Dodds and D. J. Watts, "Universal behavior in a generalized model of contagion," *Physical Review Letters*, vol. 92, no. 21, article 218701, 2004.

[31] D. J. Watts, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002.

[32] P. S. Dodds and D. J. Watts, "A generalized model of social and biological contagion," *Journal of Theoretical Biology*, vol. 232, no. 4, pp. 587–604, 2005.

[33] W. Wang, X.-L. Chen, and L.-F. Zhong, "Social contagions with heterogeneous credibility," *Physica A: Statistical Mechanics and Its Applications*, vol. 503, pp. 604–610, 2018.

[34] J. C. Miller, "A note on a paper by Erik Volz: sir dynamics in random networks," *Journal of Mathematical Biology*, vol. 62, no. 3, pp. 349–358, 2011.

[35] W. Wang, M. Tang, P. Shu, and Z. Wang, "Dynamics of social contagions with heterogeneous adoption thresholds: crossover phenomena in phase transition," *New Journal of Physics*, vol. 18, no. 1, article 013029, 2016.

[36] W. Wang, M. Tang, H.-F. Zhang, H. Gao, Y. Do, and Z.-H. Liu, "Epidemic spreading on complex networks with general degree and weight distributions," *Physical Review E*, vol. 90, no. 4, article 042803, 2014.

[37] P. Erdos and A. Rényi, "On random graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[38] A. Hagberg, D. Schult, P. Swart et al., *Networkx. High productivity software for complex networks*, Webová Strá Nka, 2013, https://networkx.lanl.gov/wiki.