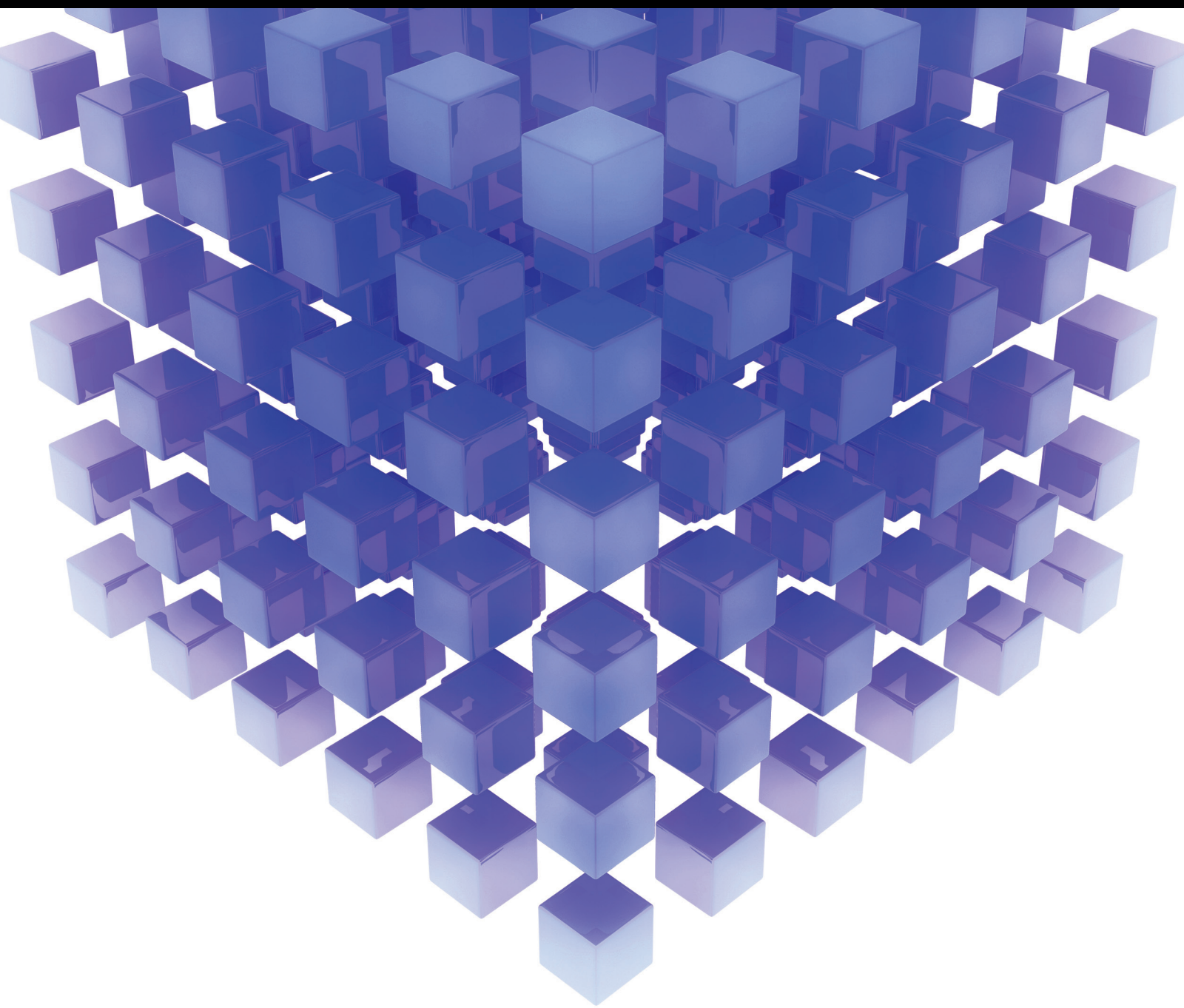
Mathematical Problems in Engineering

Security and Privacy Protection of Social Networks in Big Data Era

Lead Guest Editor: Lixiang Li
Guest Editors: Zonghua Zhang, Kaoru Ota, and Liu Yuhong



Security and Privacy Protection of Social Networks in Big Data Era

Mathematical Problems in Engineering

Security and Privacy Protection of Social Networks in Big Data Era

Lead Guest Editor: Lixiang Li
Guest Editors: Zonghua Zhang, Kaoru Ota, and Liu Yuhong

Editorial Board

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 6872587, 2 pages
https://doi.org/10.1155/2018/6872587

Editorial

Security and Privacy Protection of Social Networks in Big Data Era

Lixiang Li,[1] **Kaoru Ota**,[2] **Zonghua Zhang**,[3] **and Yuhong Liu**[4]

[1] *Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Muroran Institute of Technology, Hokkaido, Japan*
[3] *Telecom Lille, Villeneuve d'Ascq, France*
[4] *Santa Clara University, Santa Clara, CA, USA*

Correspondence should be addressed to Lixiang Li; li_lixiang2006@163.com

Received 13 December 2017; Accepted 14 December 2017; Published 3 January 2018

Big Data draws the attention not only because of its great power but for the severe security and privacy challenges it brings. With the sources from various formats of user generated contents like digital video, blogging, forms, online social conversations, and so on, Big Data can be a strong tool to serve the users as well as attacking them. With the increasing applications of Big Data, profit-driven attacks are emerging rapidly, raising great challenges for data security, privacy, and trust. Hence, the recent research focus on Big Data Era has more emphasis on the protection of security and privacy. As the existence of the contradiction between large quantity of data with various formats and limited bandwidth and storage and computation power, the current defense solutions cannot resolve the problem entirely. So the conventional security mechanisms for small-scale or isomorphic data should be modified to adapt to the exponential increment of user generated data. It is important to develop new lightweight cryptographic algorithms (protocols), data mining, data organization and data optimization models, and performance evaluation methods to protect the security and the privacy of Big Data.

This special issue involves 14 original papers selected by the editors so as to present the most significant results in the above-mentioned topics. These papers are organized as follows.

Two papers on attribute-based encryption and revocation are as follows: "Modified Ciphertext-Policy Attribute-Based Encryption Scheme with Efficient Revocation for PHR System," by H. Zheng et al.; "Research on Ciphertext-Policy Attribute-Based Encryption with Attribute Level User Revocation in Cloud Storage," by G. Wang and J. Wang.

Five papers on user preference matching, classification model, and community discovery are as follows: "SHMF: Interest Prediction Model with Social Hub Matrix Factorization," by C. Cui et al.; "A Quick Negative Selection Algorithm for One-Class Classification in Big Data Era," by F. Zhu et al.; "Multiview Community Discovery Algorithm via Nonnegative Factorization Matrix in Heterogeneous Networks," by W. Tao and L. Yang; "A Stable-Matching-Based User Linking Method with User Preference Order," by X. Wang et al.; "New Collaborative Filtering Algorithms Based on SVD++ and Differential Privacy," by Z. Xian et al.

Five papers on attack and intrusion detection/handle are as follows: "Economic Levers for Mitigating Interest Flooding Attack in Named Data Networking," by L. Wang et al.; "A Universal High-Performance Correlation Analysis Detection Model and Algorithm for Network Intrusion Detection System," by H. Zhu et al.; "Games Based Study of Nonblind Confrontation," by Y. Yang et al.; "An Effective Conversation-Based Botnet Detection Method," by R. Chen et al.; "Identifying APT Malware Domain Based on Mobile DNS Logging," by W. Niu et al.

Two papers on data transmission are as follows: "Semitensor Product Compressive Sensing for Big Data Transmission in Wireless Sensor Networks," by H. Peng et al.; "Efficient Data Transmission Based on a Scalar Chaotic Drive-Response System," by A. Li and C. Wang.

Acknowledgments

We would like to thank all authors who submitted their works for this special issue. Lixiang Li is supported by the National Key Research and Development Program of China (Grant no. 2016YFB0800602) and the National Natural Science Foundation of China (Grant no. 61573067).

Lixiang Li
Kaoru Ota
Zonghua Zhang
Yuhong Liu

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6808190, 10 pages
https://doi.org/10.1155/2017/6808190

Research Article

Modified Ciphertext-Policy Attribute-Based Encryption Scheme with Efficient Revocation for PHR System

Hongying Zheng,[1] Jieming Wu,[2] Bo Wang,[2] and Jianyong Chen[2]

[1]*School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen, China*
[2]*School of Computer and Software Engineering, Shenzhen University, Shenzhen, China*

Correspondence should be addressed to Jianyong Chen; jychen@szu.edu.cn

Received 26 January 2017; Accepted 3 August 2017; Published 30 August 2017

Academic Editor: Haipeng Peng

Attribute-based encryption (ABE) is considered a promising technique for cloud storage where multiple accessors may read the same file. For storage system with specific personal health record (PHR), we propose a modified ciphertext-policy attribute-based encryption scheme with expressive and flexible access policy for public domains. Our scheme supports multiauthority scenario, in which the authorities work independently without an authentication center. For attribute revocation, it can generate different update parameters for different accessors to effectively resist both accessor collusion and authority collusion. Moreover, a blacklist mechanism is designed to resist role-based collusion. Simulations show that the proposed scheme can achieve better performance with less storage occupation, computation assumption, and revocation cost compared with other schemes.

1. Introduction

Personal health record (PHR) system is a novel application that can bring great convenience in healthcare. The privacy and security of PHR are the major concerns of the users, which could hinder further development and wide adoption of the system [1, 2]. PHR is a typical usage of cloud storage, taking advantages of elastic computing resources to provide flexible, pervasive, and on-demand health cloud service. Patients store their PHRs in cloud storage servers and therefore can share these data with friends or doctors conveniently. However, such promising cloud-based application meets new security challenges: (1) Since PHRs need to be shared among doctors, researchers, patients, and so on, the sharing scenario is complicated. Patients should be able to control the access in a fine-grained manner. (2) PHRs may be migrated among different cloud storage servers which cannot be fully trusted. Therefore, patients cannot rely on servers to protect their PHRs. Traditionally, outsourced data is usually encrypted with cipher-key and the storage servers are responsible for distributing cipher-keys to legal accessors. However, such mechanism is just secure in specific domain, but not suitable for PHR system which works across several domains.

It is significant to find out a fine-grained access control technique for PHR system. In recent years, attribute-based encryption (ABE) [3–8] seemed to be a promising technique for such one-file-multiaccess cloud storage scenario. In ABE algorithm, patient can control the security by directly specifying access policies for their outsourced PHRs, while the third-party entities, named authorities, are responsible for attribute management and key distribution. Cloud storage only needs to store the encrypted PHRs. In this way, PHR service is oriented to patients across several domains.

Typically, ABE schemes work in two models, key-policy ABE (KP-ABE) [9] and ciphertext-policy ABE (CP-ABE) [10]. KP-ABE applies policy in attribute keys of accessors. Therefore, once a key is predefined and is used to encrypt PHRs, accessors who can decrypt them are limited. Accessor can only decrypt the PHRs associated with a set of attributes that satisfies the key. That is to say, PHR owner should know all attributes that accessors own before he encrypts one PHR, so that he can associate a correct set of attributes. It is not natural and practical, unless the attributes of accessors are generated and distributed by PHR owner himself. CP-ABE scheme works in the opposite manner, which is conceptually

closer to the traditional access control methods, such as Role-Based Access Control (RBAC) [10]. The access policy is set by PHR owner during PHR encryption, where the policy is a Boolean formula consisting of public attributes and logical operations, like "AND" and "OR." PHR owner does not need to know who can access his PHRs because it is responsibility of authority. Only the accessors with attributes that satisfy access policy can decrypt ciphertext of PHR. Evidently, it is more reasonable to implement CP-ABE scheme in public attributes scenario, and it is also convenient for PHR owner without keeping online all the time.

Based on the application scenarios of KP-ABE and CP-ABE, Li et al. [11] proposed a PHR system framework that combines KP-ABE and CP-ABE together. In the framework, users are divided into personal domains (PSDs) and public domains (PUDs) according to their roles. Usually, PHR owners (patients) normally knows users who access the system via PSDs. It would be better to apply revocable KP-ABE scheme for PSDs [12], so that patients are responsible for defining attributes and authorizing accessors. Professional users access the system via PUDs. They should have public roles, such as doctor and researcher. Therefore, it is better for the attributes in PUD to be defined and authorized by third-party attribute authorities (abbreviated as AA_s in this paper). Li et al. uses Chase-Chow multiauthority ABE scheme (CC MA-ABE) [13] with an attribute revocation method to control the attributes in PUDs.

Although there are some advantages for the division of user domains, several shortcomings still exist for Li's ABE scheme [11] (abbreviated as Li's MA-ABE), which are listed as follows: (1) Since it works based on CC MA-ABE which is exactly a variant KP-ABE scheme, it is limited on a strict "AND" policy over a predetermined set of authorities. As commented by Lewko and Waters [14], such policy is not flexible and expressive. In order to get the same function of CP-ABE, it uses an additional conjunctive normal form (CNF) rule for generation of both policy and encryption. (2) PUDs and PSDs have to apply different ABE schemes and work in parallel. However, our paper reveals an implicit collusion, named role-based collusion, between users from PUDs and PSDs. Specifically, users in PSDs may also have professional roles, such as doctors with public attributes in PUDs. In this situation, one PHR owner can prevent specific accessor from PSD by associating his PHR with a set of PSD attributes but may fail to prevent this accessor from accessing via PUD. For example, patient A has a friend B who works as a physician in hospital C. Patient A goes to hospital C for diagnosis. He specifies an access policy for his encrypted PHR to allow all the physicians in hospital C access. However, he suddenly remembers that his friend B also works there and he does not want him to know the diagnosis. Although patient A does not authorize friend B to decrypt via PSD, he cannot stop friend B from accessing via PUD.

There exist several MA-CP-ABE schemes [11, 13, 15–18], but they are not designed for PUD's scenario. Commented by paper [14], CC MA-ABE [13, 17] is limited by the strict "AND" policy. Muller et al. proposed an ABE scheme that can realize any access structure but needs an authentication center [16]. The usage of authentication center may face security and performance bottleneck because all the authorities should be controlled by center. Lin et al. [15] gave a scheme without authentication center but needs to fix the set of authorities ahead of time. It can resist collusion of users less than m, where m is a chosen parameter at setup phase. Lewko's ABE solution [14] is flexible but lacks attribute revocation mechanism. Ruj et al. proposed a solution based on Lewko's ABE to make attribute revocable [19]. However, it requires PHR owner to stay online for revocation and its efficiency is quite low.

More importantly, the role-based collusion which is significant for PHR system is not solved in these previous MA CP-ABE schemes. In order to resist the collusion, our proposed MA CP-ABE scheme designs a blacklist for owner. Each user (PHR owner) can specify a blacklist of accessor identities that cannot decrypt his data from PUD. This blacklist is delegated to a third-party authority that the owner trusts. The authority tags each blacklist with a unique public attribute in PUD, so that the owner can use this unique public attribute to specify his access policy. However, the amount of public attributes will increase linearly with PUD users, which results in a heavy burden for authorities.

Consequently, our paper aims to construct the CP-ABE scheme for PUD scenario which has efficient revocation and supports multiple authorities without an authentication center. Compared with Li's ABE scheme in PUD, our proposed scheme realizes access control with flexible access policy. Moreover, the proposed role-based collusion is also solved efficiently. Our contributions are concluded as follows.

(1) We propose a modified multiauthority CP-ABE scheme based on Lewko's scheme [14]. With it, PHR owner can specify flexible and expressive access policy to protect their outsourced PHRs. Meanwhile, authorities need not communicate with each other or be controlled by an authentication center. The number of attributes is almost unrestricted since the increase of attributes does not occupy more resources.

(2) We proposed an efficient attribute revocation mechanism for our scheme. Attribute can be revoked efficiently through the proxy reencryption and lazy revocation, while the scheme does not need an authentication center and any additional communications among authorities.

(3) To resist the role-based collusion, we suggest a blacklist solution to prevent it. By replacing the specific attribute master key and public key with hash value of attribute's descriptive name, the storages in authorities keep small even when number of attributes increases.

2. Related Work

Sahai and Waters [8] proposed the first ABE scheme, in which ciphertext is encrypted and associated with a set α of attributes. An accessor can successfully decrypt ciphertext if and only if he gets a set β of attributes components where the set overlap between the two attributes sets, that is, $|\alpha \cap \beta|$, is beyond a predefined threshold. Afterwards, Goyal et al.

[9] proposed KP-ABE scheme, in which a set of attributes from an accessor is constructed through a tree-like policy which is taken as key of the accessor. The leaf nodes of the tree associated with attributes and the nonleaf nodes are logical operations, such as "or" and "and." Data owner associates his ciphertext with a set of attributes. Once the associated attributes satisfy a specific key-policy of accessor, the accessor can decrypt the ciphertext. However, the data owner should know all the keys of accessors before he encrypts the data and then he can suitably associate the ciphertext with corresponding attributes. Such requirements of KP-ABE are not suitable for public access scenario, where the data owner cannot predict which person can access his data.

Consequently, Bethencourt et al. [10] proposed CP-ABE which is conceptually closer to the traditional access control methods, such as RBAC. CP-ABE scheme attaches access policy in ciphertext instead of attributes of accessors. It is more intuitive for the data owner to specify such policy at the time he encrypts the data. For accessors, they should own enough attributes issued by the third party, named authorities, to decrypt the ciphertext correctly. Furthermore, ordered binary decision diagram (OBDD) is used to describe access policies in CP-ABE. The system makes full use of both the powerful description ability and the high calculating efficiency of OBDD and improve both performance and efficiency [20]. However, only one single authority may cause bottleneck of performance [21]. Moreover, it is more natural and practical with multiple professional organizations (authorities) to manage distinct sets of attributes. Security can be improved with the multiauthority because an attacker should compromise several authorities at the same time to get the keys associated with enough sets of attributes for decryption.

There are already some attempts to solve multiauthority ABE problem with new cryptographic solutions. Chase and Chow [13] firstly proposed a multiauthority ABE scheme (CC MA-ABE) in which each user is authorized based on a global identifier (GID), such as a social security number. The GID plays a linchpin to associate users' keys from different authorities together. But the solution still relies on an authentication center and the access policy is not flexible and expressive which is limited on "AND" gate policy over the predetermined set of authorities. Later, Li et al. [11] proposed an ABE scheme with attribute revocation mechanism based on CC MA-ABE, which is limited on a rule of CNF in the access policy. A threshold multiauthority CP-ABE access control scheme was proposed for public cloud storage with which both security and performance are improved [22].

Actually, it is important for MA CP-ABE to support an expressive and flexible access policy. For example, American Medical Association (AMA) authorizes attributes of medical professional licenses, such as junior nurse license and experienced nurse license, while American Hospital Association (AHA) authorizes attributes of affiliations, such as hospital A and hospital B. If one patient thinks that the diagnosis and treatment in hospital A are better than those in hospital B, he may specify an access policy that permits the nurses with any level of license in hospital A to access his PHR files, and only allow the nurses with junior level of license from hospital B access. Such expressive policy is presented as policy = ((/junior nurse level/ $\vee$ /experienced nurse level/) $\wedge$ /hospital A/) $\vee$ (/junior nurse level/ $\wedge$ /hospital B/). The policy can be transformed to the "AND" policy; for example, policy = $\{(A_1 = a_{1,1}) \vee \cdots \vee (A_1 = a_{1,d_1})\} \wedge \cdots \wedge \{(A_m = a_{m,1}) \vee \cdots \vee (A_m = a_{m,d_m})\}$, where A_m refers to the mth authority and a_{m,d_i} refers to the policy managed by A_m and one authority has only one clause [11].

There are some other schemes which can set the access policy in any Boolean formula over attributes from any number of authorities. Among them, Muller proposed another MA-ABE scheme which is realized on any access structure with an authentication center. Yang and Jia [18] proposed a variant CP-ABE scheme to support multiauthority, but it still requires an additional authentication center to generate user secret key and authority secret key. Moreover, it is weak in revocation security. Based on Yang's scheme, an extensive scheme was proposed to withstand the vulnerability [23]. For MA-ABE scheme with an authentication center to control multiple authorities, once the authentication center is broken, the entire ABE system will be compromised. Therefore, it should be fully trusted which is hard to guarantee. Moreover, the whole ABE system is hard to be expanded. Some researches try to remove the authentication center from MA CP-ABE schemes. Chase and Chow [13] used pseudorandom functions (PRFs) between different authorities without the center. However, it is still limited on "AND" access policy over a determined set of authorities. Lin et al. [15] proposed a threshold based ABE scheme that is decentralized and enforces an efficient attribute revocation scheme. The system is collusion-resistant for fewer m users, where m is chosen statically during the setup phase. However, the authorities set should be configured before the setup phase and is fixed in running. The authorities should interact with each other at the setup phase and the access policy is inflexible. Later, Lewko and Waters [14] proposed a scheme for decentralized ABE scenario, in which the authorities work independently without coordination among them. A main drawback is that the scheme has no revocation function. Although a further paper (DACC) [19] addressed it, the computations of key update and communication overhead for attribute revocation are quite heavy. Besides, DACC requires the data owner to take part in revocation and transmit an updated ciphertext component to every unrevoked user. It means that the data owner should keep being online all the time, as is unreasonable in practical application scenario.

Attribute revocation is an important issue for an ABE system and benefits security of the system. Once a malicious user is identified by an authority, all his attributes or one of his specific attributes should be revoked by the authority, which means the malicious user can no longer decrypt the ABE-generated ciphertext associated with those attributes. In single authority ABE scheme, Yu et al. [7] introduced the concept of proxy reencryption into CP-ABE to realize attribute revocation, in which the affected attribute components of ciphertext and the attributes components stored in terminals of unrevoked users are updated via reencryption. Inspired by paper [7], Yang and Jia [18] proposed the CP-ABE scheme

Table 1: Comparison among previous MA CP-ABE schemes and ours.

	Lin [15]	Muller [16]	Chase [17]	Lewko [14]	DACC [19]	Li [11]	Yang [18]	Ours
Flexible access policy	√	√		√	√		√	√
Resistance of accessor collusion		√	√	√	√	√	√	√
Without an authentication center	√		√	√	√	√		√
Authority independence	√	√		√	√	√	√	√
Efficient revocation			√	—		√	√	√

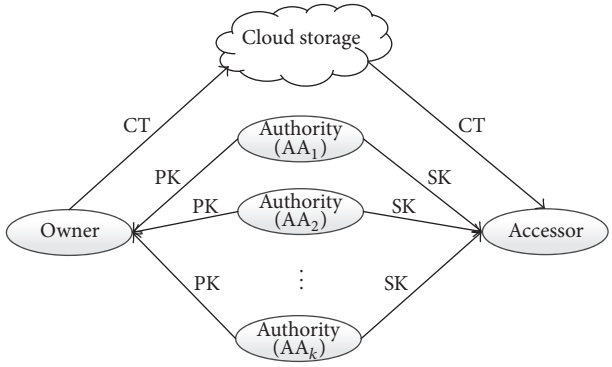


Figure 1: MA CP-ABE system model.

with a more efficient revocation than that in [19]. However, it requires an authentication center to control the multiple authorities. Based on the above depiction, the comparisons among previous MA CP-ABE schemes and our proposed scheme are listed in Table 1.

3. System Model and Security Definition for MA CP-ABE

3.1. System Model. The MA CP-ABE scheme for PUD involves three kinds of participants, that is, cloud storage, authorities, and users (including data owner and accessors), as shown in Figure 1. The scheme consists of five basic algorithms: *System Setup*, *Authority Setup*, *Encrypt*, *KeyGen*, and *Decrypt*. They are described as follows.

System Setup $(\lambda) \rightarrow (para)$. The setup algorithm takes security parameter λ as input and outputs global parameters *para*.

Authority Setup $(para) \rightarrow (msk, pk)$. Each attribute authority (AA) runs its own authority setup process. The setup algorithm takes system global parameters *para* and AA's descriptive attributes as input. Then, for each attribute that AA manages, AA generates a master key *msk* and the corresponding public key *pk*. The master keys msk_s are kept secret, while the public keys pk_s are published.

Encrypt $(D, para, policy\ \mathcal{T}, pk_s) \rightarrow (CT = \{D^{/}, policy\ \mathcal{T}, pAC_s\})$. Once the data owner gets public keys pk_s from authorities, he can execute encryption process in his own terminal. The algorithm takes pk_s from several authorities, data D for encryption, and an access policy $\mathcal{T}$ specified by the data owner as inputs. Then, the algorithm encrypts D to a ciphertext $D^{/}$ and generates a public attribute component (abbreviated as pAC) for each leaf node of $\mathcal{T}$. The whole data tuple of $CT = \{D^{/}, policy\ \mathcal{T}, \mathrm{pAC}_s\}$ is the final ciphertext tuple and is uploaded to cloud storage.

KeyGen $(para, msk) \rightarrow (SK : \{uAC_s\})$. Each authority manages its own attributes set and is responsible for key distribution to legal users (accessors). Once an authority authenticates identity of an accessor, it will process key generation which takes the master keys mk_s for a requested set of attributes $\omega^{/}$ as input and outputs user attribute components (abbreviated as uAC_s) for each attribute. All the attributes uAC_s generated for the specific accessor are collected as secret key of the accessor SK and sent back to the accessor secretly.

Decrypt $(para, CT, SK_s, pk_s) \rightarrow (M)$. An accessor executes the decryption algorithm which takes the ciphertext tuple CT from cloud storage and the public keys pk_s and secret keys SK_s from authorities as inputs. If the attributes set associated with SK_s satisfies access policy $\mathcal{T}$, the accessor can decrypt the plaintext data M. Otherwise, it returns an error symbol $\perp$.

3.2. PHR Upload and Access. Based on CP-ABE scheme (Figure 1), we can easily figure out the PHR upload and PHR access procedures. Specifically, once a data owner needs to upload his specific PHR file *"pFile"* to cloud storage, he does the following steps: (1) Cut the data into contents segments *s*. (2) Pick random content key *ck* for each content segment. (3) Encrypt the segment via symmetric cryptography and get result $s^{/} = E_{ck}(s)$. (4) Define an access policy over a set of attributes, encrypt content key *ck* as owner data *M* via our proposed MA CP-ABE scheme, and get the ciphertext tuple CT. (6) Finally upload $s^{/}$ and CT together as an integrated tuple to the cloud storage. The data owner can go offline and authorities perform other key distribution workflows.

When an accessor needs to read the plaintext of one specific PHR on the cloud storage, he should process the following steps: (1) Get the whole ciphertext tuple $s^{/}$ and CT from the cloud storage. (2) Read the access policy from the CT and know a minimal set of attributes required for decryption. (3) Get identity authenticated by several authorities, with which these authorities can return the keys associated with attributes (uAC_s) to the accessor, respectively. (4) Collect enough keys to recover content key *ck* from CT. (5) Decrypt $s^{/}$ to *s* via symmetric cryptography by content key *ck* and then construct the original PHR file *"pFile."*

4. Modified MA CP-ABE Scheme for PUD

4.1. Scheme Construction. Our proposed MA CP-ABE scheme has five algorithms, that is, *System Setup, Authority Setup, KeyGen, Encrypt, and Decrypt*. They are depicted as follows.

System Setup $\rightarrow$ (*para*). System first selects a bilinear group $\mathbb{G}$ of order $N = p_1p_2p_3$ and bilinear map function $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ and then picks a generator g_1 of $\mathbb{G}_{p_1}$ [14, 24]. A hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$ is used to map global identities GID_s of an accessor and descriptive names of his attributes, such as *doctor*, to elements in $\mathbb{G}$. Once the hash function is fixed, the value $H(\mathrm{GID})$ is modelled as a random oracle. Finally, all these system parameters are published as $para = (\hat{e}, g_1, H(\cdot), N)$.

Authority Setup $(para) \rightarrow (msk, pk)$. For each authority AA_k which manages attributes set $\mathbb{A}_k$, AA_k takes *para* as input and generates two public keys as $g_1^{\alpha_k}$, $g_1^{\beta_k}$ where the two values α_k, β_k are picked randomly from $\mathbb{Z}_N$. The values $msk_k = (\alpha_k, \beta_k)$ are stored secretly by AA_k as master keys, while the public keys $pk_k = (g_1^{\alpha_k}, g_1^{\beta_k})$ are published.

KeyGen $(para, msk) \rightarrow (SK = \{uAC_s\})$. Suppose that a legal accessor with *GID* requests authority AA_k for attributes set A_u and he owns attributes set $A_{u,k}$ in AA_k. Then AA_k will generate secret key (SK) of the accessor which is associated with attributes set $A_u \cap A_{u,k}$. Specifically, for each attribute $i \in A_u \cap A_{u,k}$, AA_k generates a user attribute component $(\mathrm{uAC}_i = H(i)^{\alpha_k} \cdot H(\mathrm{GID})^{\beta_k})$ for the accessor. Finally, all the components $\{\mathrm{uAC}_i\}|_{i \in A_u \cap A_{u,k}}$ are combined as secret keys of the accessor and $\mathrm{SK} = \{\mathrm{uACs}\}$ is sent back to the accessor secretly for further decryption.

Encrypt $(D, para, policy\ \mathcal{T}, pk_s) \rightarrow (CT = \{D^{/}, policy\ \mathcal{T}, pAC_s\})$. In encryption phase, the data owner specifies an access policy tree $\mathcal{T}$ to restrict the accessors. The encryption algorithm encrypts data D into $D^{/} = D \cdot \hat{e}(g_1, g_1)^s$, where the value $s \in \mathbb{Z}_n$ is selected randomly. Meanwhile, a set of public attribute components (pAC_s) will be generated according to the value s and the access policy $\mathcal{T}$.

Specifically, as shown in previous paper [11], any monotone access tree $\mathcal{T}$ can be translated to an access structure $(\mathscr{M}, \rho)$ over the involved attributes, where $\mathscr{M}$ is a $\ell \times n$ matrix and ℓ denotes the number of leaf nodes in the access tree $\mathcal{T}$. The function ρ maps the xth row of matrix $\mathscr{M}_x$ to an attribute $i = \rho(\mathscr{M}_x)$. The encryption algorithm chooses two random vectors $\overrightarrow{v} = (s, r_2, \ldots, r_n) \in \mathbb{Z}_N^n$ and $\overrightarrow{v'} = (0, r_2', \ldots, r_n') \in \mathbb{Z}_N^n$ and then computes $\lambda_x = \overrightarrow{v} \cdot \mathscr{M}_x$ and $\nu_x = \overrightarrow{v'} \cdot \mathscr{M}_x$. Notice that the former vector $\overrightarrow{v}$ is used to distribute the value s, while the latter vector formula distributes the zero value 0. For each leaf node x of $\mathcal{T}$ associated with attribute $i = \rho(\mathscr{M}_x)$, the algorithm computes the three pAC_s as follows, where the value μ_x is picked arbitrarily in $\mathbb{Z}_n$

$$\begin{aligned} \mathrm{pAC}_{0\,(x,i)} &= \hat{e}(g_1, g_1)^{\lambda_x} \cdot \hat{e}(H(i), g_1)^{\alpha_k \cdot \mu_x}, \\ \mathrm{pAC}_{1\,(x,i)} &= g_1^{\mu_x}, \\ \mathrm{pAC}_{2\,(x,i)} &= g_1^{\beta_k \cdot \mu_x + \omega_x}. \end{aligned} \tag{1}$$

Finally, the owner sends the ciphertext $D^{/}$ together with pAC_s and access structure $(\mathscr{M}, \rho)$ to the semitrust cloud storage. The uploaded data CT is presented as

$$\begin{aligned} CT = \Big(D^{/}, (\mathscr{M}, \rho), \{\mathrm{pAC}_{0(x,i)}, \mathrm{pAC}_{1(x,i)}, \mathrm{pAC}_{2(x,i)}\} \mid \\ (i = \rho(\mathscr{M}_x)) \,\&\, (1 \le x \le R)\Big). \end{aligned} \tag{2}$$

Decrypt $(para, CT, SK_s, pk_s) \rightarrow (D)$. An accessor receives CT from the cloud storage, finds out the minimal set of attributes $\mathbb{A}_u$ for decryption according to the policy $\mathcal{T}$, and then requests corresponding AA_s for attributes (uAC_s). Notice that the minimal attributes set $\mathbb{A}_u$ is mapped to ℓ' rows of matrix $\mathscr{M}$. The rows set is labeled as $\{I_x\}$, where $|\{I_x\}| = \ell'$ and $\ell' \le \ell$. According to submatrix $\{I_x\}$, the algorithm can compute ℓ' values $\{\zeta_x \in \mathbb{Z}_n\}|_{x \in \{I_x\}}$, which has the relationship with $s = \sum_{x \in \{I_x\}} \zeta_x \lambda_x$ and $0 = \sum_{x \in \{I_x\}} \zeta_x \omega \omega_x$ *(interpolation)*.

Consequently, for each leaf node which is associated with the xth row of $\{I_x\}$, the algorithm can decrypt it via the following formula:

$$\begin{aligned} &\frac{\mathrm{pAC}_{0\_(x,i)} \cdot \hat{e}\left(H(\mathrm{GID}), \mathrm{pAC}_{2_{x,i}}\right)}{\hat{e}\left(\mathrm{uAC}_i, \mathrm{pAC}_{1_{x,i}}\right)} \\ &= \frac{\hat{e}(g_1, g_2)^{\lambda_x} \cdot \hat{e}(H(i), g_1)^{\alpha_k \cdot \mu_x} \cdot \hat{e}\left(H(\mathrm{GID}), g_1^{\beta_k \cdot \mu_x + \omega_x}\right)}{\hat{e}\left(H(i)^{\alpha_k} \cdot H(\mathrm{GID})^{\beta_k}, g_1^{\mu_x}\right)} \\ &= \hat{e}(g_1, g_1)^{\lambda_x} \cdot \hat{e}(H(\mathrm{GID}), g_1)^{\omega_x}. \end{aligned} \tag{3}$$

By collecting ℓ' decryption values of leaf nodes, the algorithm can easily recover value $\hat{e}(g_1, g_1)^s$ via interpolation depicted as follows:

$$\begin{aligned} &\prod_{x \in \{I_x\}} \left(\hat{e}(g_1, g_1)^{\lambda_x} \cdot \hat{e}(H(\mathrm{GID}), g_1)^{\omega_x}\right)^{\zeta_x} \\ &= \hat{e}(g_1, g_1)^{\sum_{x \in \{I_x\}} (\zeta_x \cdot \lambda_x)} \cdot \hat{e}(H(\mathrm{GID}), g_1)^{\sum_{x \in \{I_x\}} (\zeta_x \cdot \omega_x)} \\ &= \hat{e}(g_1, g_1)^s \cdot \hat{e}(H(\mathrm{GID}), g_1)^0 = \hat{e}(g_1, g_1)^s. \end{aligned} \tag{4}$$

Finally, the plaintext D is computed by $D = D'/\hat{e}(g_1, g_1)^s$.

4.2. Efficient Lazy Revocation. There are two levels of revocation, that is, attribute revocation and accessor revocation. The attribute revocation is done by updating the attribute associated pACs stored in cloud storage, so that the previous authenticated pACs is no longer useful for decryption. The accessor revocation can be done by revocation of all the attributes that an accessor owns.

Normally, the command of attribute revocation is started from authority when there are changes in management of accessors. Firstly, authority AA_k sends update parameter to

the cloud storage and then the cloud storage updates pAC_s via proxy reencryption technique [12]. In our revocation scheme, the corresponding pAC_s will not be updated until someone requests them. Specifically, the cloud storage stores the update parameters in an attribute history list (AHL) for each attribute revocation command. Once a ciphertext (associated with a set of pAC_s) is requested, it can be updated only once according to AHL, although the update parameters have been updated many times and recorded in AHL. Such mechanism is called lazy revocation, which can accumulate update of parameters over time. Our revocation model is more efficient than DACC's solution [19] when AA_k delegates most computation workloads to the cloud storage and the lazy revocation is used.

For accessors, once pAC_s stored in the cloud storage is updated, their corresponding uAC_s can no longer decrypt the ciphertext. Consequently, these accessors need to request authorities to update parameters. Instead of regenerating the accessors' uAC_s, the authorities can simply generate parameters, that is, update keys (UK_s), and let these accessors update their uAC_s at their terminal.

In previous papers [11, 12, 25], the revocation methods will generate the same update keys for all accessors. This is efficient but weak in security. Therefore, our proposed revocation scheme can support two methods. One method is to generate the same update parameters for all accessors, and the other one is to generate different update parameters for different accessors. It is obvious that the former method is efficient but has potential risk in some circumstance. The latter method is the opposite. PHR system can choose either method according to its strategy and environment.

Attribute Revocation $(para, msk) \rightarrow (UK_{aAC}, UK_{pAC})$. To execute the revocation command for attribute i, its corresponding authority AA_k takes public system parameters *para* and its own master key (α_k, β_k) as input. Then AA_k generates regeneration key UK_{pAC} for the cloud storage and generates UK_{aAC} for the accessors. All these regeneration keys are transmitted secretly.

Method 1 (Same Update Parameter). Specifically, AA_k selects a random value $\alpha' \in Z_N$ and then generates $UK_{aAC_i} = UK_{pAC_i} = H(i)^{\alpha'_k - \alpha_k}$. The cloud storage updates the attribute i associated $pAC_{0\_(x,i)}$ through (5). uAC_i of the accessor is updated through (6) at the terminals of accessors or at the authority

$$\begin{aligned} pAc'_{0\_(x,i)} &= pAC_{0\_(x,i)} \cdot \hat{e}\left(UK_{pAC_i}, pAC_{1_{x,i}}\right) \\ &= \hat{e}\left(g_1, g_1\right)^{\lambda_x} \cdot \hat{e}\left(H\left(i\right), g_1\right)^{\alpha'_k \cdot \mu_x}, \end{aligned} \tag{5}$$

$$uAC'_i = uAC_i \cdot UK_{aAC_i} = H\left(i\right)^{\alpha'_k} \cdot H\left(\mathrm{GID}\right)^{\beta_k}. \tag{6}$$

Method 2 (Different Update Parameters). Specifically, AA_k selects random values $\alpha'_k, \beta'_k \in \mathbb{Z}_n$ and generates $UKpACi = H(i)^{\alpha'_k - \alpha_k}$ and $UKpACi = \beta'_k - \beta k$ for the cloud storage. For each accessor with GID, AA_k generates specific $UKaACi, GID = H(i)^{\alpha'_k - \alpha_k} \cdot H(\mathrm{GID})^{\beta'_k - \beta_k}$. The cloud storage updates the attribute i associated $pAC0\_(x,i)$ and $pAC2\_(x,i)$ through (7) and (8). The accessor's $uACi$ is updated through (9)

$$\begin{aligned} pAC'_{0\_(x,i)} &= pAC_{0\_(x,i)} \cdot \hat{e}\left(UK_{pAC_i}, pAC_{1_{x,i}}\right) \\ &= \hat{e}\left(g_1, g_1\right)^{\lambda_x} \cdot \hat{e}\left(H\left(i\right), g_1\right)^{\alpha'_k \cdot \mu_x} \end{aligned} \tag{7}$$

$$pAC'_{2\_(x,i)} = pAC_{2\_(x,i)} \cdot PAC_{1_{x,i}}^{UK_{pAC_i}} = g_1^{\beta'_k \cdot \mu_x + \omega_x} \tag{8}$$

$$uAC'_i = uAC_i \cdot UK_{\alpha AC_i, \mathrm{GID}} = H\left(i\right)^{\alpha'_k} \cdot H\left(\mathrm{GID}\right)^{\beta'_k}. \tag{9}$$

Accessor Revocation. Supposing that the attributes set $\mathbb{A}_\alpha$ is owned by the accessor, the corresponding authority AA_k can execute attribute revocations for these $|\mathbb{A}_\alpha|$ attributes in total. Moreover, to avoid fake revocation commands, both the authority and the cloud storage use digital signature technique to confirm validity as implemented in paper [12].

4.3. Collusion Resistant. The same as most of previous papers [11, 18], our proposed MA CP-ABE scheme can resist both accessor collusion and authority collusion. Besides, the malicious but implicit role-based collusion can also be resisted.

As discussed in Introduction, role-based collusion is caused by the fact that PHR owner cannot predict the exact user identity who is an accessor from PUD because the attribute authentication is controlled by the third authority party. To resist the collusion, it is essential for PHR owner to specify a blacklist, which contains the access identities that are not allowed access from PUD and delegates the blacklist to a third authority party. The authority maps each blacklist to an attribute, such as attribute "*Alice*$'$*s Blacklist*1," so that an owner can combine such attributes in his access policy in PUD to restrict specific identity from access. Normally, the amount of blacklist attributes will grow linearly with users in PHR system. Fortunately, our proposed ABE construction is efficient in managing attributes because the algorithms replace attribute master keys with the hash values of attributes' descriptive names. The storage for attribute management can keep small at the authority even when the number of attributes increases. It means that the blacklist solution is highly efficient.

Accessor collusion denotes that different accessors will combine their attribute components (pACs) together for decryption of a file despite the fact that they do not have enough attributes to decrypt it alone. Our proposed MA CP-ABE scheme can resist the accessor collusion by embedding the accessor's hash value into their pACs. Consequently, the temporary result in decryption phase, that is, $\hat{e}(g1, g1)\lambda x \cdot \hat{e}(H(\mathrm{GID}), g1)\mu x$, differs among accessors. Therefore, the decryption process is resisted.

Authority collusion is an important security metric in multiauthority scenario. In our proposed scheme, since the authorities do not communicate with each other or have no predefined parameters among them, the authority collusion is impossible in our proposed scheme.

Table 2: Storage overhead on each entity.

	DACC	Yang	Ours
Authority	$2 * n_{\text{att}}$	$n_{\text{att}} + 2 * n_{\text{user}} + 3$	2
Owner	$n_c + 2 * n_{\text{att}} + 2$	$3 * n_{\text{AA}} + 2 * n_{\text{att}} + 3$	$2 * n_{\text{AA}} + 1$
Accessor	$n_{\text{pAC}_s} + n_{\text{att}}$	$2 * n_{\text{AA}} + n_{\text{att}} + 2$	n_{att}
Cloud storage	$(3 * \text{avg} + 1) * n_{\text{cipher}}$	$(4 * \text{avg} + 3) * n_{\text{cipher}}$	$(3 * \text{avg} + 1) * n_{\text{cipher}}$

Table 3: Time consumption of different types of operation.

Type	Description	Time for 1000 operations
$T0$	Time for two-vector multiplication	Depending on the vector length
$T1$	Time for one PBC pairing operation	875443 (us)
$T2$	Time for one PBC exponent operation	1419140 (us)
$T3$	Time for one PBC multiply operation	13264 (us)
$T4$	Time for one PBC addition operation	1196 (us)

Table 4: Computation efficiency.

	Time for encryption	Time for decryption
DACC	$n_{\text{pAC}_s} \cdot (2 \cdot T0 + 5 \cdot T2 + 2 \cdot T3) + (T2 + T3)$	$n^{/} \cdot (2 \cdot T1 + T2 + 3 \cdot T3)\text{pAC}_s$
Yang	$n_{\text{pAC}_s} \cdot (T0 + 5 \cdot T2 + 2 \cdot T3) + (3 \cdot T2 + n_{\text{AA}} \cdot T3)$	$n \cdot (4 \cdot T1 + 2 \cdot T2 + 4 \cdot T3) + n_{\text{AA}} \cdot (2 \cdot T1 + T3) + (T2 + T3)\text{pAC}_s$
Ours	$n_{\text{pAC}_s} \cdot (2 \cdot T0 + T1 + 4 \cdot T2 + 2 \cdot T3) + (T2 + T3)$	$n^{/} \cdot (2 \cdot T1 + T2 + 3 \cdot T3) \cdot n_{\text{pAC}_s}$

5. Performance

In this section, we will compare performances between our proposed scheme and previous MA CP-ABE schemes in aspects of storage cost, computation efficiency, and revocation cost. Since Li's ABE scheme for PUD is actually a variant KP-ABE scheme, we will compare our scheme with both DACC's [19] and Yang's scheme [18].

5.1. Storage. The storage overheads on each entity are listed in Table 2. Notice that n_{user} is the amount of users (accessors) in PHR system, n_{att} denotes the number of all attributes, n_{AA} denotes the number of authorities, n_{cipher} is the number of all ciphertext tuples n_c stored in cloud storage, and n_{pAC_s} denotes the number of generated pAC_s at terminal of accessor. For comparison, the storage overheads of these parameters are n_c, n_{cipher}, n_{user}, and $n_{\text{pAC}_s} > n_{\text{att}} > n_{\text{AA}}$. Specifically, storage overhead at authority (AA) is mainly the space occupation of master keys and public keys for attributes. Since our proposed scheme uses hash values to replace keys for attributes, the storage space at authorities can be saved evidently. We suppose that each ciphertext is associated with avg attributes on average. From Table 2, it is evident that our scheme has the smallest storage overhead at authority, terminal of owner, terminal of accessor, and cloud storage compared with both DACC's and Yang's schemes.

5.2. Computation Efficiency. In this section, we compare the computation costs for these three schemes by implementing them on a Linux system with an Intel Core i7 CPU at 2.20 GHz and 1.00 GB RAM. The codes are constructed based on the Pairing-Based Cryptography (PBC) library version 0.5.14. A symmetric elliptic curve α-curve whose base field size is 512 bits is set up to execute the pairing operation. The group order of α-curve is of 160 bits; that is, p_1 is a 160-bit length prime. All the simulation results come from the average of 20 trials.

Before the simulations, time consumption values of four PBC functional operations are compared which are listed in Table 3. It is obvious that pairing operation and exponent operation consume more time than multiplication and addition. Furthermore, time consumption for encryption and decryption is shown in Table 4 where n' denotes the number of pACs required in each decryption.

We compare the computation efficiencies of both encryption and decryption in two criteria: (1) The number of authorities is changeable while the number of attributes in each authority is fixed. (2) The number of authorities is fixed while the number of attributes in each authority is changeable. The result is shown in Figure 2. In the first simulation, the number of related authorities (x-axis) changes from 2 to 20, and the involved attributes of each authority are set to be 10. Time for encryption is shown in Figure 2(a), while time for decryption is presented in Figure 2(b). The second simulation is the opposite. The number of involved attributes in each authority changes from 2 to 20, and related authorities are set to be 10. Time for encryption and time for decryption are shown in Figures 2(c) and 2(d), respectively. Evidently, our proposed scheme has better performance in computation efficiency because of less number of PBC exponent operations.

5.3. Revocation Cost. As shown in Table 5, we use expressions to denote the communication overheads between terminals and the cloud storage. In DACC, it is the responsibility

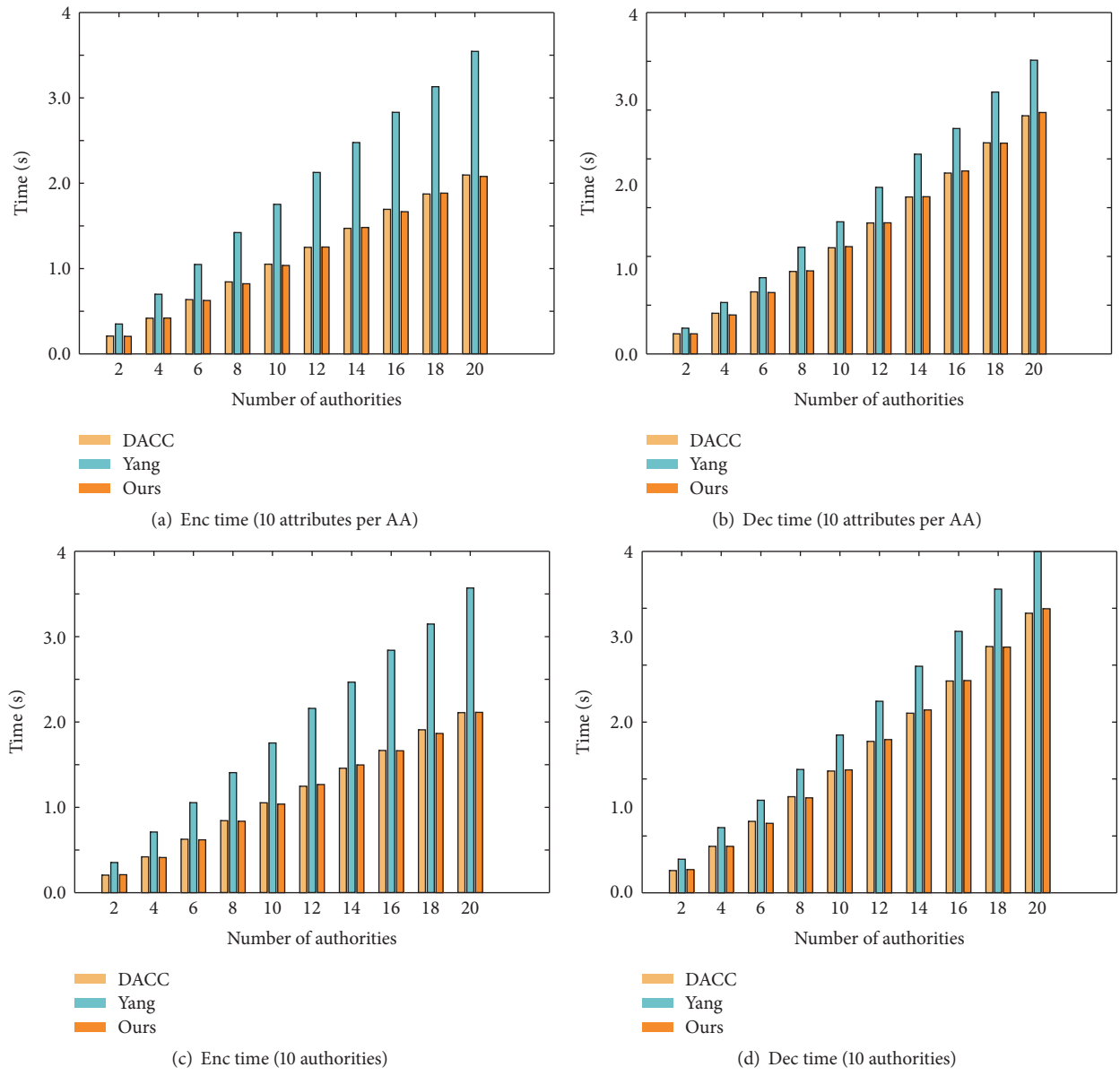


Figure 2: Time for encryption (Enc time) and decryption (Dec time).

Table 5: Communication overhead of attribute revocation.

	DACC	Yang's scheme	Ours (method 1)	Ours (method 2)
Update parameters for accessors	$(n'_{\mathrm{pAC}_s} * n'_{\mathrm{user}} + 1) * \lvert p_1 \rvert$	$n'_{\mathrm{user}} * \lvert p_1 \rvert$	$n'_{\mathrm{user}} * \lvert p_1 \rvert$	$n'_{\mathrm{user}} * \lvert p_1 \rvert$
Update parameters for cloud storage server	$n'_{\mathrm{pAC}_s} * \lvert p_1 \rvert$	$2 * \lvert p_1 \rvert$	$\lvert p_1 \rvert$	$2 * \lvert p_1 \rvert$

Notes. n'_{pAC_s} is the number of ciphertexts which is associated with the revoked attribute i. n'_{user} is the number of unrevoked accessors. $\lvert p_1 \rvert$ is the length of each update parameter.

of data owner to generate update parameters for attribute revocation. In some other schemes, authority generates the update parameters and the data owner can stay offline. It is clear that DACC is inefficient because the data owner should regenerate all the related pACs manually. Both Yang's scheme and our two revocation methods (the same update parameters and different update parameters) use the proxy reencryption technique to reduce communication cost and computation cost.

Time revocation for different number of attributes is shown in Figure 3 where the x-axis denotes number of the revoked attributes and the y-axis is time consumption. For

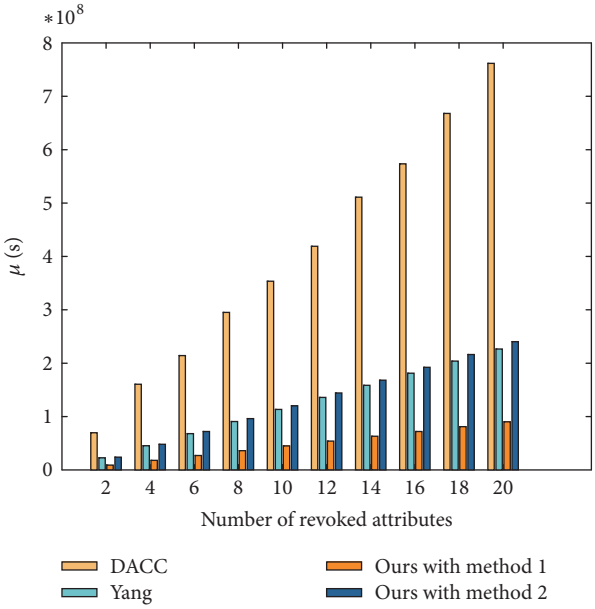


Figure 3: Revocation time with different number of attributes.

simplify, we set the related ciphertext as n tuples and each ciphertext is associated with 10 attributes (so that $n'_{\mathrm{pAC}_s} = 1000 * 10$).

It is inefficient for the data owner to generate update parameters for each attribute associated pAC in DACC, which means the data owner should always keep being online. Our second revocation method (different update parameters) is as efficient as Yang's scheme [18], while our first revocation method (same update parameter) is more efficient because it generates the same update parameters for all accessors. It is noticed that the difference of computation time will be more obvious if n'_{pAC_s} or n'_{user} are getting bigger. From both Table 5 and Figure 3, we can conclude that our scheme has higher efficiency in in communication and computation.

6. Conclusion

In this paper, we proposed a modified MA CP-ABE scheme to implement fine-grained access control. Our proposed scheme supports expressive access policy and can resist user collusion without an authentication center. Moreover, two types of attribute revocation methods, which can revoke attribute efficiently, are proposed. The system can choose one of them according to different application scenarios. Simulations and analysis show that the proposed scheme can achieve less in storage occupation, computation assumption, and revocation cost compared with other schemes.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 61402291 and the Technology Planning Project from Guangdong Province, China, under Grant no. 2014B010118005.

References

[1] J. Li, "Ensuring privacy in a personal health record system," *Computer*, vol. 48, no. 2, Article ID 7042698, pp. 24–31, 2015.

[2] Y. Yang and M. Ma, "Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 746–759, 2016.

[3] A. Ge, J. Zhang, R. Zhang, C. Ma, and Z. Zhang, "Security analysis of a privacy-preserving decentralized key-policy attribute-based encryption scheme," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2319–2321, 2013.

[4] M. Li, "Fractal time series—a tutorial review," *Mathematical Problems in Engineering*, Article ID 157264, Art. ID 157264, 26 pages, 2010.

[5] M. Li, "Record length requirement of long-range dependent teletraffic," *Physica A. Statistical Mechanics and its Applications*, vol. 472, pp. 164–187, 2017.

[6] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265–1277, 2016.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communication Security, (ASIACCS '10)*, pp. 261–270, April 2010.

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in cryptology*, vol. 3494 of *Lecture Notes in Comput. Sci.*, pp. 457–473, Springer, Berlin, 2005.

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

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

[11] 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.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, March 2010.

[13] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 121–130, Chicago, Ill, USA, November 2009.

[14] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in cryptology*, vol. 6632 of *Lecture Notes in Comput. Sci.*, pp. 568–588, Springer, Heidelberg, 2011.

[15] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Information Sciences. An International Journal*, vol. 180, no. 13, pp. 2618–2632, 2010.

[16] S. Muller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Information security and cryptology*, vol. 5461 of *Lecture Notes in Comput. Sci.*, pp. 20–36, Springer, Berlin, 2009.

[17] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 515–534, Springer, Berlin, Germany, 2007.

[18] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, 2014.

[19] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: distributed access control in clouds," in *Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '11)*, pp. 91–98, Changsha, China, November 2011.

[20] L. Li, T. L. Gu, L. Chang, Z. B. Xu, Y. N. Liu, and J. Y. Qian, "A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram," *IEEE Access*, vol. 5, pp. 1137–1145, 2017.

[21] L. Ibraimi, M. Asim, and M. Petković, "Secure management of personal health records by applying attribute-based encryption," in *Proceedings of the 6th International Workshop on Wearable, Micro, and Nano Technologies for Personalized Health*, pp. 71–74, Oslo, Norway, June 2009.

[22] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484–1496, 2016.

[23] X. Wu, R. Jiang, and B. Bhargava, "On the security of data access control for multiauthority cloud storage systems," *IEEE Transactions on Services Computing*, vol. PP, no. 99, 2015.

[24] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of cryptography*, vol. 3378 of *Lecture Notes in Comput. Sci.*, pp. 325–341, Springer, Berlin, 2005.

[25] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-Based Data Sharing Scheme Revisited in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1661–1673, 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1383891, 12 pages
https://doi.org/10.1155/2017/1383891

Research Article

SHMF: Interest Prediction Model with Social Hub Matrix Factorization

Chaoyuan Cui,[1] Hongze Wang,[2] Yun Wu,[3] Sen Gao,[4] and Shu Yan[1]

[1]*Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, Anhui 230031, China*
[2]*University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*Institute of Applied Technology, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, Anhui 230088, China*
[4]*University of Science and Technology of China, Hefei, Anhui 230031, China*

Correspondence should be addressed to Shu Yan; yanshu@iim.ac.cn

Received 24 January 2017; Accepted 5 June 2017; Published 22 August 2017

Academic Editor: Zonghua Zhang

With the development of social networks, microblog has become the major social communication tool. There is a lot of valuable information such as personal preference, public opinion, and marketing in microblog. Consequently, research on user interest prediction in microblog has a positive practical significance. In fact, how to extract information associated with user interest orientation from the constantly updated blog posts is not so easy. Existing prediction approaches based on probabilistic factor analysis use blog posts published by user to predict user interest. However, these methods are not very effective for the users who post less but browse more. In this paper, we propose a new prediction model, which is called SHMF, using social hub matrix factorization. SHMF constructs the interest prediction model by combining the information of blogs posts published by both user and direct neighbors in user's social hub. Our proposed model predicts user interest by integrating user's historical behavior and temporal factor as well as user's friendships, thus achieving accurate forecasts of user's future interests. The experimental results on Sina Weibo show the efficiency and effectiveness of our proposed model.

1. Introduction

Online microblog systems such as Sina Weibo, Twitter, and Facebook provide a convenient platform for users to share their information. The number of such social media users showed exponential growth in last decade. A recent snapshot of the friendship network Facebook indicated that there are over 1 billion users in it. These social networks are becoming not only effective means to connect their friends but also powerful information dissemination and marketing platforms to spread ideas, fads, and political opinions.

Microblog contains a vast amount of information, and topics of users and user groups always change with hotspot at home and abroad or over time. In this context, research on user interest prediction is useful in network marketing, public opinion analysis, or even public security [1]. Generally, interest prediction is to generate potential and possible topics in the next time point according to one's historical blog posts. Unfortunately, blog posts are almost short text; both user-keyword matrix and user-topic matrix of microblogs are relatively very sparse. Moreover, in the prediction model, contents of the related matrices transfer with lots of factors, such as time information and friendship in social hub. Therefore, interest prediction is still a challenging problem.

It should be noted that user interest prediction is different from user interest detection, as the latter mainly focuses on mining users' current interests. Interest prediction remains a relatively understudied problem that poses two main challenges. First, user interest in microblog changes over time or time interval. In the time-aware prediction model, user's temporal preference is an important aspect. Furthermore, long-term preference and short-term preference will result in different prediction result. Second, user interest is a dynamic phenomenon; it maybe migrates due to the topic migration of one's social hub. In the real world, capturing user's friendship and their topics is difficult.

Recently, a lot of models for prediction have been investigated [2–4]. A typical method exploits the probabilistic

matrix factorization (PMF) technique to learn latent features for users and topics. These kinds of algorithms are mostly based on the blog posts published by user to predict his interest.

In fact, we observed several interesting phenomena. There exist some users who publish less but browse more blog posts and we call them silent type users. Such users may have very explicit interest and just may be prudent to express their ideas. And they do publish their opinion at an appropriate moment. However, existing prediction models always fail to predict their interests. Another kind of users expands their social hubs by focusing on new friends' topics they are interested in. We call them interactive type users. In other words, the interest of such users can be represented by the interest of direct neighbors in their social hubs to some extent. Obviously, prediction models ignoring the impact of this interactive property always result in incomplete forecast.

In order to overcome the shortcomings of existing works, combining our observations about microblog, this paper proposes a social hub matrix factorization-based model for user interest prediction model in microblog, which is called SHMF. SHMF incorporates the impact of user's social hub on user's interests in our model to improve the quality of prediction. The experimental results on Sina Weibo dataset show that our approach improves the prediction accuracy and the performance efficiency.

The rest of this paper is organized as follows. The related work is discussed in Section 2. Some preliminary knowledge and research are introduced in Section 3. We present our proposed model in Section 4 and give the implementation details in Section 5. In Section 6, we describe the real datasets we used in our experiments. Our experiments are reported in Section 7. Finally, we conclude the paper and present some directions for future work in Section 8.

2. Related Work

With regard to user interest prediction in microblog, there are a series of mature methods that are based on probability matrix factorization of probabilistic graph model. Probabilistic graph model is a kind of model which can concisely express complex probability distribution, effectively calculate the edge and condition distribution, and conveniently learn the parameters and hyperparameters in probability model [5], while probability matrix factorization based on this model is often used to predict the user's interests and recommendations.

In 2008, Salakhutdinov and Mnih [2] proposed a probability matrix factorization (PMF) method for the traditional collaborative filtering algorithm which cannot solve the problem of the recommendation of large sparse dataset and cold start. Experiments on datasets of Netflix demonstrate the effectiveness of PMFs on large number of sparse unbalanced datasets. In the same year, Ma et al. [3] applied PMF to social network and socialization recommendation and analyzed the complexity and prediction accuracy of this method in detail. In 2010, combining the characteristics of social networks, Jamali and Ester [4] proposed a social probability matrix factorization (SocialMF) model based on the consideration of the social trust relationship between users. This model promotes the application prospect of PMF in socialization recommendation. In 2003, Sun et al. [6] proposed a method to model the user's timing behavior and combined this method with the SocialMF to predict the Weibo user's interest, the experimental results of which prove that this way of modeling is more effective than the traditional recommendation algorithm based on label information. Taking into account the fact that user interest is changing over time, Bao et al. [7] introduced a new temporal and social PMF-based (TS-PMF) method to predict users' interests in microblog. Compared with previous methods of interest prediction, this method has higher accuracy.

The above studies neglect the impact of the information of the blogs posted by others in their social hub on the user's future interest and behavior, when they establish the Weibo user interest prediction model. Aiming at this problem, in this paper, we propose a new user interest prediction model (SHMF) based on PMF, which combines user's history behavior, user's social trust relationship, and the impact of the information of the users' social hub on the user's interests in the future. And it designs experiments on the Sina microblog real dataset to prove that this prediction model and the algorithm of the model are superior to the previous prediction model in top-n accuracy [8].

3. Preliminaries

In this section, we give the notations that will be used in the following discussions. In prediction model, we have a set of users $\{u_1, u_2, \ldots, u_n\}$ and a set of topics $\{v_1, v_2, \ldots, v_m\}$ in a microblog dataset.

The users' interests expressed by user-topic matrix are given in $R \in R^{n*m}$, where $r_{ij} = 1$ if user u_i has published posts on topic v_j. We divide users' historical data into N time points $(T_1, T_2, \ldots, T_t)$ and construct a set of user-topic matrix $\mathbb{R}_1 = \{R_{11}, R_{12}, \ldots, R_{1t}\}$ to represent user's interests over time. Furthermore, considering the impact of user's social hub on his/her interest, we can construct a set of user's social hub-topic matrix $\mathbb{R}_2 = \{R_{21}, R_{22}, \ldots, R_{2t}\}$ according to the blogs posted by friends of his/her social hub.

In microblog, each user can follow others whom he is interested in; then users' friendships can be described as a user-user matrix $F_1 \in R^{n*n}$, where $F_{1,ij} = 1$ which denotes that u_i has followed u_j. Each user can mainly read the blogs posted by his friends of his social hub. Obviously, there are interactions among different users' social hubs. Users' social hubs can be described as a hub-hub matrix $F_2 \in R^{n*n}$. We set $F_{2,ij} = n_{ij}/n_i$ if the number of users in the intersection of hub α_i and hub α_j is n_{ij} and the number of users in hub α_i is n_i. Hub α_i is a set of users who are followed by u_i, and we have a set of user social hubs $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

Generally, user interest prediction model is to generate a user-interest matrix in the next time segment. The basic matrix factorization (MF) approach finds the approximate matrix of the original matrix in the low-rank space as a predictive approximation matrix. It has been proven to be effective to learn the latent characteristics of users and topics

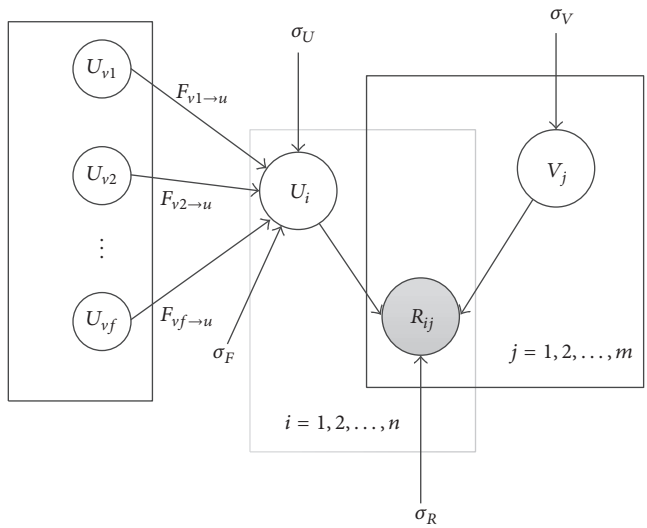


Figure 1: Graphical model of SocialMF.

and predict the scores using these latent characteristics. The conditional probability of the known scores is defined as

$$P\left(R \mid U, V, \sigma_R^2\right) = \prod_{i=1}^{n}\prod_{j=1}^{m}\left[N\left(r_{ij} \mid g\left(U_i^T V_j\right), \sigma_R^2\right)\right]^{I_{ij}^R}. \quad (1)$$

As is shown in (1), $U \in R^{d*n}$ and $V \in R^{d*m}$ are the latent characteristics of users and topic feature matrices, with column vectors U_i and V_j representing d-dimensional user-latent and topic-latent feature vectors, respectively; $r_{ij} \approx U_i^T V_j$, where U_i^T is the transpose of U_i. $N(x \mid \mu, \sigma^2)$ is the Gaussian distribution with mean μ and variance σ^2, and I_{ij}^R is the indicator function that is equal to 1 if $r_{ij} = 1$ and is equal to 0 otherwise. The function $g(x)$ is a logistic function with the formula $g(x) = 1/(1 + \exp(-x))$, which makes it possible to bound x within the range $[0, 1]$.

In fact, the relations among users in social network architecture play an important role in users' behaviors [9, 10]. Specifically, a user is more and more similar to his/her friends. SocialMF model incorporates social influence into the MF approach for prediction, adding the user-user relationship matrix $F \in R^{n*n}$:

$$\begin{aligned} P\left(U \mid F, \sigma_U^2, \sigma_F^2\right) &\propto P\left(U \mid \sigma_U^2\right) * P\left(U \mid F, \sigma_F^2\right) \\ &= \prod_{i=1}^{n} N\left(U_i \mid 0, \sigma_U^2 I\right) \\ &\quad * \prod_{i=1}^{n} N\left(U_i \mid \sum_j F_{ij} U_j, \sigma_F^2 I\right). \end{aligned} \quad (2)$$

Figure 1 shows the graphical model corresponding to (2). In Figure 1, the edges among the latent feature vectors of users are representatives of the trust relationship among users and the degree of trust of user u on user v is $F_{u\to v}$.

The user-topic matrices in PMF and SocialMF model are all constructed from the user's historical behavior information and do not take time influence into account. Meanwhile TS-PMF model incorporates characteristics of the user interest over time and adds the exponential decay function to analyze the user-topic matrices [7]. TS-PMF is designed to utilize users' sequential interest matrices $\{R_{11}, R_{12}, \ldots, R_{1t}\}$ and the users' friendships matrix $F \in R^{n*n}$ to predict users' interest in the near future. In time t, the conditional distribution probability of the observed items in R_{1t} is similar to that in (1):

$$\begin{aligned} &P\left(R_t \mid U_t, V_t, \sigma_{R_t}^2\right) \\ &\quad = \prod_{i=1}^{n}\prod_{j=1}^{m}\left[N\left(R_{tij} \mid g\left(U_{t,j}^T V_{t,j}\right), \sigma_{R_t}^2\right)\right]^{I_{ij}^{R_t}}. \end{aligned} \quad (3)$$

Adding the exponential decay function to analyze the change of user interest, the computing formulation is listed as follows:

$$\begin{aligned} M_{U_t} &= \theta \sum_{k=1}^{t-1} \exp\left(\frac{t-k}{\beta}\right) U_k, \\ M_{V_t} &= \theta \sum_{k=1}^{t-1} \exp\left(\frac{t-k}{\beta}\right) V_k. \end{aligned} \quad (4)$$

The user's latent feature vector is affected by his historical interests and his friends' interests. Therefore, the conditional distribution probability of users' latent features can be expressed like this:

$$\begin{aligned} &P\left(U_t \mid \{R_1, R_2, \ldots, R_{t-1}\}, F, \sigma_{U_t}^2, \sigma_{F_t}^2\right) \\ &\quad \propto P\left(U_t \mid \{R_1, R_2, \ldots, R_{t-1}\}, \sigma_{U_t}^2\right) * P\left(U_t \mid F, \sigma_F^2\right) \\ &\quad = \prod_{i=1}^{n} N\left(U_{t,i} \mid M_{U_{i,j}}, \sigma_{U_t}^2 I\right) \\ &\qquad * \prod_{i=1}^{n} N\left[U_i \mid \sum_j F_{ij} U_j, \sigma_F^2 I\right]. \end{aligned} \quad (5)$$

Now, through a Bayesian inference, we have the following equation for the posterior probability over latent features of users and topics:

$$\begin{aligned} &P\left(U_t, V_t \mid \{R_1, R_2, \ldots, R_t\}, F, \sigma_{U_t}^2, \sigma_{V_t}^2, \sigma_F^2, \sigma_{R_t}^2\right) \\ &\quad \propto P\left(R_t \mid U_t, V_t, \sigma_{R_t}^2\right) \\ &\qquad * P\left(U_t \mid \{R_1, R_2, \ldots, R_{t-1}\}, \sigma_{U_t}^2\right) \\ &\qquad * P\left(U_t \mid F, \sigma_F^2\right) \\ &\qquad * P\left(V_t \mid \{R_1, R_2, \ldots, R_{t-1}\}, \sigma_{V_t}^2\right). \end{aligned} \tag{6}$$

Maximizing the log of the posterior distribution with regard to U_t and V_t is equivalent to minimizing the following sum-of-squared-errors objective function (we can find a local optimal value of the objective function by performing gradient descent):

$$\begin{aligned} &E\left(U_t, V_t \mid \{R_1, R_2, \ldots, R_t\}, F\right) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}^{R_t}\left(R_{t,ij} - g\left(U_{t,j}^T V_{t,j}\right)\right)^2 \\ &\quad + \frac{\lambda_{U_t}}{2}\left\|U_t - M_{U_t}\right\|_F^2 + \frac{\lambda_{V_t}}{2}\left\|U_t - M_{V_t}\right\|_F^2 \\ &\quad + \frac{\lambda_F}{2}\sum_{i=1}^{n}\left(U_{t,i} - \sum_v F_{iv}U_{t,v}\right)^T\left(U_{t,i} - \sum_v F_{iv}U_{t,v}\right). \end{aligned} \tag{7}$$

4. Social Hub User Interest Prediction Model

In this section, we present our model, SHMF, to incorporate impact of user's social hub into MF approach for prediction. SHMF combines user's historical behavior, social trust relationship, and blog articles posted by friends in user's social hub.

Independence Hypothesis. Information of blogs posted in users' social hub influences users' interests independently.

Based on the above hypothesis, we have

$$\begin{aligned} &P\left(R_{1t}, R_{2t} \mid U_t, V_t, \sigma_{R_{1t}}^2, \sigma_{R_{2t}}^2\right) \\ &\quad \propto P\left(R_{1t} \mid U_t, V_t, \sigma_{R_{1t}}^2\right) * P\left(R_{2t} \mid U_t, V_t, \sigma_{R_{2t}}^2\right) \\ &\quad = \prod_{i=1}^{n}\prod_{j=1}^{m}\left[N\left(R_{1t,ij} \mid g\left(U_{t,i}^T V_{t,j}\right), \sigma_{R_{1t}}^2\right)\right]^{I_{ij}^{R^{1t}}} \\ &\qquad * \prod_{i=1}^{n}\prod_{j=1}^{m}\left[N\left(R_{2t,ij} \mid g\left(U_{t,i}^T V_{t,j}\right), \sigma_{R_{2t}}^2\right)\right]^{I_{ij}^{R^{2t}}}. \end{aligned} \tag{8}$$

Therefore, the conditional distribution probability of users' latent features can be expressed as follows:

$$\begin{aligned} &P\left(U_t \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_1, F_2, \right. \\ &\left.\sigma_{U_{1t}}^2, \sigma_{U_{2t}}^2, \sigma_{F_1}^2, \sigma_{F_2}^2\right) \propto P\left(U_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, F_1, \right. \\ &\left.\sigma_{U_{1t}}^2, \sigma_{F_1}^2\right) * P\left(U_t \mid \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_2, \sigma_{U_{2t}}^2, \sigma_{F_2}^2\right). \end{aligned} \tag{9}$$

Through a Bayesian inference, we have the following equation for the posterior probability over latent features of users and topics:

$$\begin{aligned} &P\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_1, \right. \\ &\left. F_2, \sigma_{U_{1t}}^2, \sigma_{U_{2t}}^2, \sigma_{V_{1t}}^2, \sigma_{V_{2t}}^2, \sigma_{F_1}^2, \sigma_{F_2}^2\right) = P\left(U_t, V_t \mid \right. \\ &\left.\{R_{11}, R_{12}, \ldots, R_{1(t)}\}, F_1, \sigma_{U_{1t}}^2, \sigma_{V_{1t}}^2, \sigma_{F_1}^2\right) * P\left(U_t, V_t \mid \right. \\ &\left.\{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_2, \sigma_{U_{2t}}^2, \sigma_{V_{2t}}^2, \sigma_{F_2}^2\right). \end{aligned} \tag{10}$$

The log of the posterior distribution for SHMF at time point t is given by

$$\begin{aligned} &\ln\left(P\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, \right. \right. \\ &\left.\left. F_1, F_2, \sigma_{U_{1t}}^2, \sigma_{U_{2t}}^2, \sigma_{V_{1t}}^2, \sigma_{V_{2t}}^2, \sigma_{F_1}^2, \sigma_{F_2}^2\right)\right) = \ln\left(P\left(U_t, V_t \mid \right. \right. \\ &\left.\left.\{R_{11}, R_{12}, \ldots, R_{1(t)}\}, F_1, \sigma_{U_{1t}}^2, \sigma_{V_{1t}}^2, \sigma_{F_1}^2\right)\right) + \ln\left(P\left(U_t, \right. \right. \\ &\left.\left. V_t \mid \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_2, \sigma_{U_{2t}}^2, \sigma_{V_{2t}}^2, \sigma_{F_2}^2\right)\right). \end{aligned} \tag{11}$$

Maximizing the log of the posterior distribution with regard to U_t and V_t is equivalent to minimizing the following sum-of-squared-errors objective function:

$$\begin{aligned} &E\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_1, \right. \\ &\left. F_2\right) = E_1\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, F_1\right) + E_2\left(U_t, \right. \\ &\left. V_t \mid \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_2\right). \end{aligned} \tag{12}$$

In (12), E_1 and E_2 can be computed by (7). It is obvious that SHMF interest prediction is actually equivalent to performing the symmetrical calculation on the loss function. Here we introduce a parameter $\lambda \in [0, 1]$ to indicate the importance of user's social hub information in user's interest. We set $\lambda = 0$ if only user's personal posting behavior is considered and set $\lambda = 1$ if only user's social hub information is considered. Thus, the loss function can be computed as follows:

$$\begin{aligned} &E\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_1, \right. \\ &\left. F_2, \lambda\right) = (1 - \lambda) * E_1\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1(t)}\}, F_1\right) \\ &\quad + \lambda * E_2\left(U_t, V_t \mid \{R_{21}, R_{22}, \ldots, R_{2(t)}\}, F_2\right), \\ &E_1\left(U_t, V_t \mid \{R_{11}, R_{12}, \ldots, R_{1t}\}, F_1\right) = \frac{1}{2} \\ &\quad \cdot \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}^{R_{1t}}\left(R_{1t,ij} - g\left(U_{t,i}^T V_{t,j}\right)\right)^2 + \frac{\lambda_{U_{1t}}}{2}\left\|U_t \right. \\ &\quad \left. - M_{U_t}\right\|_F^2 + \frac{\lambda_{V_{1t}}}{2}\left\|V_t - M_{V_t}\right\|_F^2 + \frac{\lambda_{F_1}}{2} \\ &\quad \cdot \sum_{i=1}^{n}\left(U_{t,i} - \sum_v F_{1iv}U_{t,v}\right)^T\left(U_{t,i} - \sum_v F_{1iv}U_{t,\mathrm{v}}\right), \end{aligned}$$

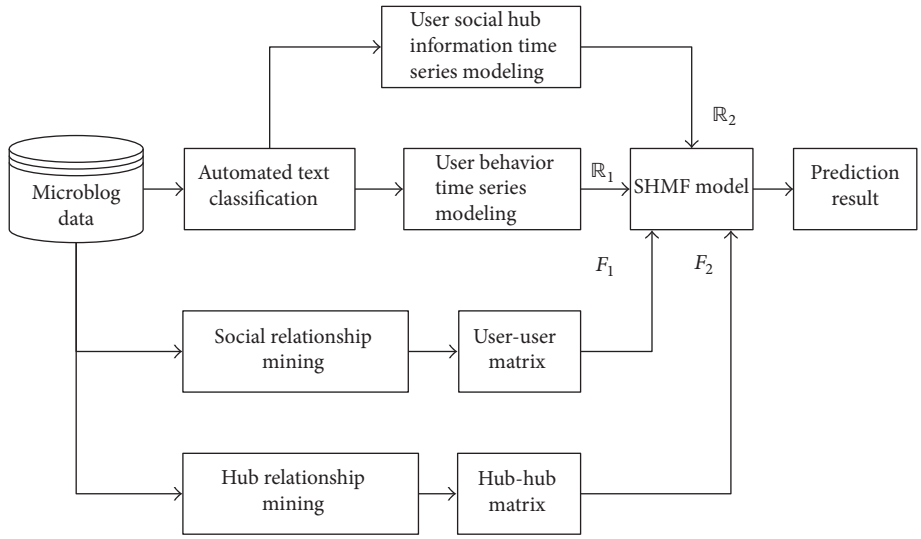


Figure 2: The framework of predicting users' interests.

$$E_2\left(U_t, V_t \mid \{R_{21}, R_{22}, \ldots, R_{2t}\}, F_2\right) = \frac{1}{2} \cdot \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}^{R_{2t}}\left(R_{2t,ij} - g\left(U_{t,i}^T V_{t,j}\right)\right)^2 + \frac{\lambda_{U_{2t}}}{2}\left\|U_t - M_{U_t}\right\|_F^2 + \frac{\lambda_{V_{2t}}}{2}\left\|V_t - M_{V_t}\right\|_F^2 + \frac{\lambda_{F_2}}{2} \cdot \sum_{i=1}^{n}\left(U_{t,i} - \sum_v F_{2iv}U_{t,v}\right)^T\left(U_{t,i} - \sum_v F_{2iv}U_{t,v}\right). \tag{13}$$

In order to reduce the computational complexity, stochastic gradient descent is used to optimize the local optimum of the loss function, as shown in (14):

$$U_t \coloneqq U_t + \alpha\left((1-\lambda)U\delta_1 + \lambda U\delta_2\right), \quad V_t \coloneqq U_t + \alpha\left((1-\lambda)V\delta_1 + \lambda V\delta_2\right), \tag{14}$$

$$U\delta_1 = I_{ij}^{R_{1t}} g'\left(U_{t,i}^T V_{t,j}\right)\left(R_{1t,ij} - g\left(U_{t,i}^T V_{t,j}\right)\right)V_{t,j} - \lambda_{U_{1t}}\left(U_{t,i} - M_{U_{t,i}}\right) - \lambda_{F_1}\left(1 - F_{1,ii}\right)\left(U_{t,i} - \sum_v F_{1,iv}U_{t,v}\right), \tag{15}$$

$$U\delta_2 = I_{ij}^{R_{2t}} g'\left(U_{t,i}^T V_{t,j}\right)\left(R_{2t,ij} - g\left(U_{t,i}^T V_{t,j}\right)\right)V_{t,j} - \lambda_{U_{2t}}\left(U_{t,i} - M_{U_{t,i}}\right) - \lambda_{F_2}\left(1 - F_{2,ii}\right)\left(U_{t,i} - \sum_v F_{2,iv}U_{t,v}\right), \tag{16}$$

$$V\delta_1 = I_{ij}^{R_{1t}} g'\left(U_{t,i}^T V_{t,j}\right)\left(R_{1t,ij} - g\left(U_{t,i}^T V_{t,j}\right)\right)U_{t,j} - \lambda_{V_{1t}}\left(V_{t,i} - M_{U_{t,i}}\right), \tag{17}$$

$$V\delta_2 = I_{ij}^{R_{2t}} g'\left(U_{t,i}^T V_{t,j}\right)\left(R_{2t,ij} - g\left(U_{t,i}^T V_{t,j}\right)\right)U_{t,j} - \lambda_{V_{2t}}\left(V_{t,i} - M_{U_{t,i}}\right), \tag{18}$$

where $g'(x) = \exp(-x)/(1+\exp(-x))^2$ is the first-order derivative of logistic function $g(x)$; $\lambda_{F_a} = \sigma_{R_{at}}^2/\sigma_{F_a}^2$, $\lambda_{U_{at}} = \sigma_{R_{at}}^2/\sigma_{U_{at}}^2$, $\lambda_{V_{at}} = \sigma_{R_{at}}^2/\sigma_{V_{at}}^2$, $\alpha = 1, 2$, and $\|\cdot\|_F^2$ are the Frobenius norm.

SHMF model provides an effective way to predict users' interests. The procedure of prediction will be described with two algorithms in Section 5. All the notations used throughout the paper are summarized in Notations.

5. Implementation

To evaluate the effectiveness and efficiency of our approach, we implemented a prototype system of user interest prediction. According to SHMF model and its variant, we provide two algorithms with different parameters and procedures.

5.1. Architecture Overview. The architecture of our implementation is illustrated in Figure 2. We first use topic model LDA to mark out topics of the microblog dataset automatically. Meanwhile, we use the sequential behaviors of users to get a set of user-topic matrices $\mathbb{R}_1 = \{R_{11}, R_{12}, \ldots, R_{1t}\}$ and a set of users' social hub-topic matrices $\mathbb{R}_2 = \{R_{21}, R_{22}, \ldots, R_{2t}\}$. Next, we capture the social relationship between users and get a user-user matrix $F_i \in R_{n*n}$ and we can get a hub-hub matrix in the same way. Finally, $\mathbb{R}_1, \mathbb{R}_2, F_1$, and F_2 are input to the SHMF model to generate the prediction result.

5.2. Algorithms. SHMF integrates user's history behavior, user's social trust relationship, and the impact of the information of user's social hub. The process of predicting users' interests with SHMF is described in Algorithm 1.

Require:
Dataset: $\{R_{11}, R_{12}, \ldots, R_{1N}\}, \{R_{21}, R_{22}, \ldots, R_{2N}\} F_1, F_2$;
The dimension of the latent feature: d;;
Parameters: $\lambda_{U_1}, \lambda_{V_1}, \lambda_{F_1}, \lambda_{U_2}, \lambda_{V_2}, \lambda_{F_2}, \theta, \beta, \lambda$;
An updating parameter: α
Convergence parameter: ε
The maximum number of iterations: K
Ensure:
The user-topic matrix in time segment $N + 1$: R_{N+1}
(1) $M_{1U_1} = \text{zeros}(d, n)$, $M_{1V_1} = \text{zeros}(d, m)$, $M_{2U_1} = \text{zeros}(d, n)$, $M_{2V_1} = \text{zeros}(d, m)$
(2) **for** $t = 1, \ldots, N$ **do**
(3) initialize $U_t, V_t : U_{t,0} = U_t$, $V_{t,0} = V_t$, $E_0 = \text{inf}$;
(4) **if** $t > 1$ **then**
(5) Compute the mean matrices M_{1U_t}, M_{1V_t}, M_{2U_t}, M_{2V_t}
(6) **end if**
(7) **for** $l = 1, \ldots, K$ **do**
(8) compute the gradient descent in Eq. (15) (16) (17) (18);
(9) updating in Eq. (14);
(10) compute E in Eq. (13);
(11) **if** $|E_0 - E| < \varepsilon$ **then**
(12) break
(13) **end if**
(14) **if** $E = \min\{E_0, E\}$ **then**
(15) $U_{t,0} = U_t$, $V_{t,0} = V_t$
(16) **end if**
(17) **end for**
(18) $U_t = U_{t,0}, V_t = V_{t,0}$
(19) **end for**
(20) predict R_{N+1} using $R_{N+1} \approx U_N^T V_N$

Algorithm 1: The process of predicting users' interests.

6. Datasets

6.1. Experimental Data. We used the dataset from 1 May 2016 to 31 May 2016, which we downloaded from Sina Weibo. This dataset includes more than 20 million microblog messages, time-stamps, and user-to-user relationships.

6.2. User Selection. The basic idea of traditional collaborative filtering is that similar users make similar choices, or similar options are chosen by similar groups of users [11]. In recent years, the basic idea of the social recommendations is gradually concerned by the researchers. The researchers of the social recommendations think that, for a social impact of consideration [12, 13], the associated users will affect each other, so the user's interest is largely influenced by the users associated with him.

Taking into account the complexity of the calculation, the selection of users is very important in the microblog user interest prediction. In a month, different users will post different numbers of microblogs. Someone only posts one, but someone posts tens of thousands. For such users who post little of microblog in a month, personal microblog information and social hub microblog information are unable to describe their interests. However, for the users who post lots of microblogs in a month, they mostly are enterprises and institutions of the official microblog or commercial procurement service, and it is meaningless to predict user's interest based on those users. To do this, we perform a statistical analysis on the dataset from Sina Weibo and find that the number of microblogs posted by most users is 100 or less as shown in Figures 3(a) and 3(b) showing histograms of the number of users with the different numbers of blog posts. In this paper, we select users who post 20 to 100 microblogs as subjects, and the number of this kind of users is about one million. After using neighbor computing [7] and stratified sampling, the 1402 users' information is selected as the experimental object.

6.3. Automatically Classify Blogs' Topics Posted by Users. After getting the user's blog information, we train the LDA model and use it to automatically classify the blogs posted by users and the blogs posted by others in user's social hub, and the number of topics is calculated by the perplexity. According to perplexity-numbers of topics curve shown in Figure 4, the best number of topics is 23 when the perplexity reached its lowest point.

7. Experiments and Analysis

In this section, effectiveness and efficiency of our SHMF model are evaluated. We conduct experiments on Intel Core i7 processor with 4 cores running at frequency of 3.60 GHz,

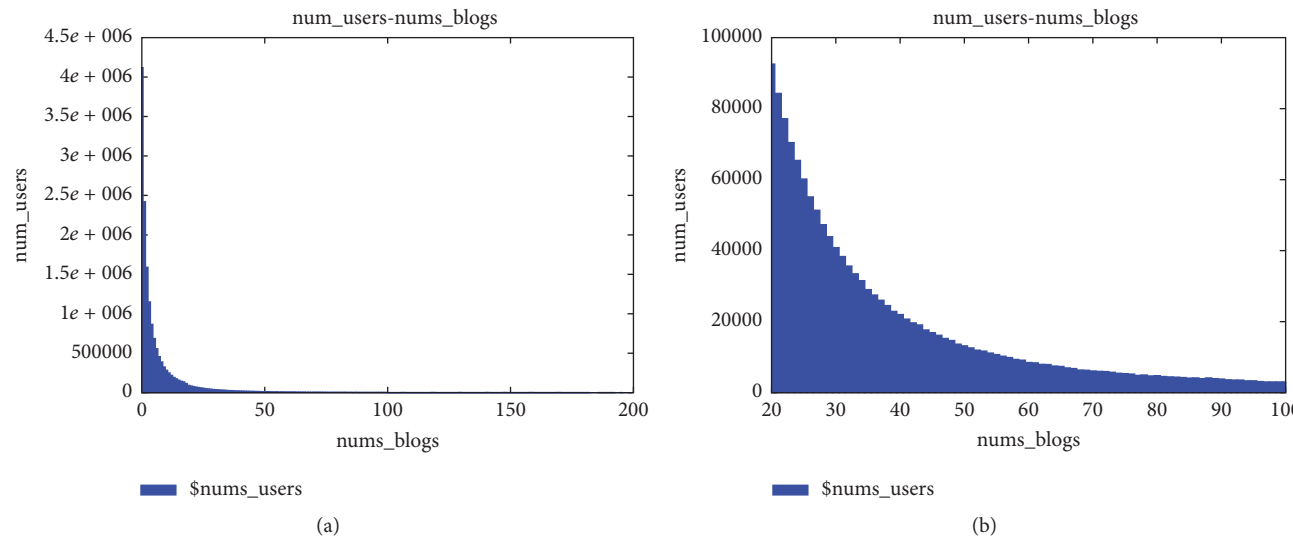


Figure 3: Statistical analysis of the dataset from Sina Weibo.

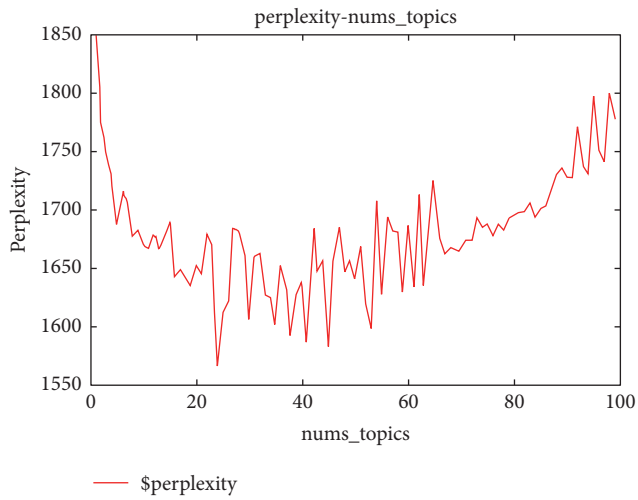


Figure 4: Perplexity-numbers of topics curve.

24 GB memory, and 1TB hard disk. The programs are run on Windows 7 Professional and Anaconda 4.1.1 (64-bit).

We first present evaluation metrics used throughout our experiments. Next, we employ the variable-controlling approach to adjust the parameters of SHMF model and the other three models. Then the prediction accuracy and the performance overhead of our model are compared with results of the other models. Finally, we will analyze the experimental results.

7.1. Metrics. Because of the great uncertainty of the behavior of user posting blogs, the recall rate has little practical significance in this issue, and in the real life users pay more attention to the top-N topic which they are most interested in. Therefore, in this paper, the precision of top-n is used as the model evaluation criteria:

$$\mathrm{Pre}_n = \frac{N_{\mathrm{correct}}(n)}{N_u \times n}. \tag{19}$$

N_u represents the number of users in the test set; and $N_{\mathrm{correct}}(n)$ represents the total number of interest topics predicted correctly in the top-n prediction results for all users in the corresponding test set.

7.2. Model Selection and Parameter Setting. We set up three experiments, PMF [2], SocialMF [4], and TS-PMF [7], as the contrastive experiments because these three methods are very often used to predict users' interests, and the three methods are in the same theoretical system as the model SHMF proposed in this paper. And then we set up an experiment for the model SHMF proposed in this paper.

First, the variable-controlling approach was used to adjust the parameters to better values, and then we compare their top-n accuracy and average accuracy.

(1) PMF Model. The PMF model has three parameters, λ_U, λ_V, d, in this paper; λ_U, λ_V are the regularization term coefficients in the loss function. The default value of λ_U, λ_V is 0.01 before setting parameters; d is the dimension of the latent features which is generally less than the rank of the original matrix. The control variable method is used to set the parameters by fixing other values and changing one. Then we can draw a graph to get the impact of each parameter on top-n accuracy. In order to reduce the computational complexity, we set $\lambda_U = \lambda_V$. The top-n accuracy varies with the parameters λ_U, λ_V, d as shown in Figure 5.

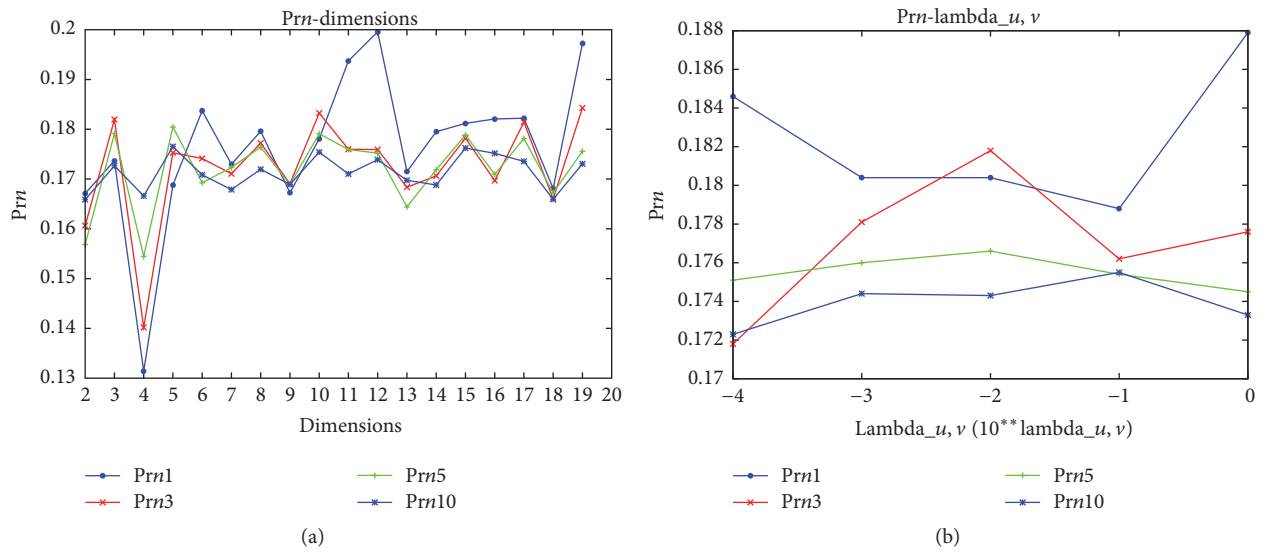


Figure 5: Impact of different values of different parameters in the PMF model on performance of user interest prediction.

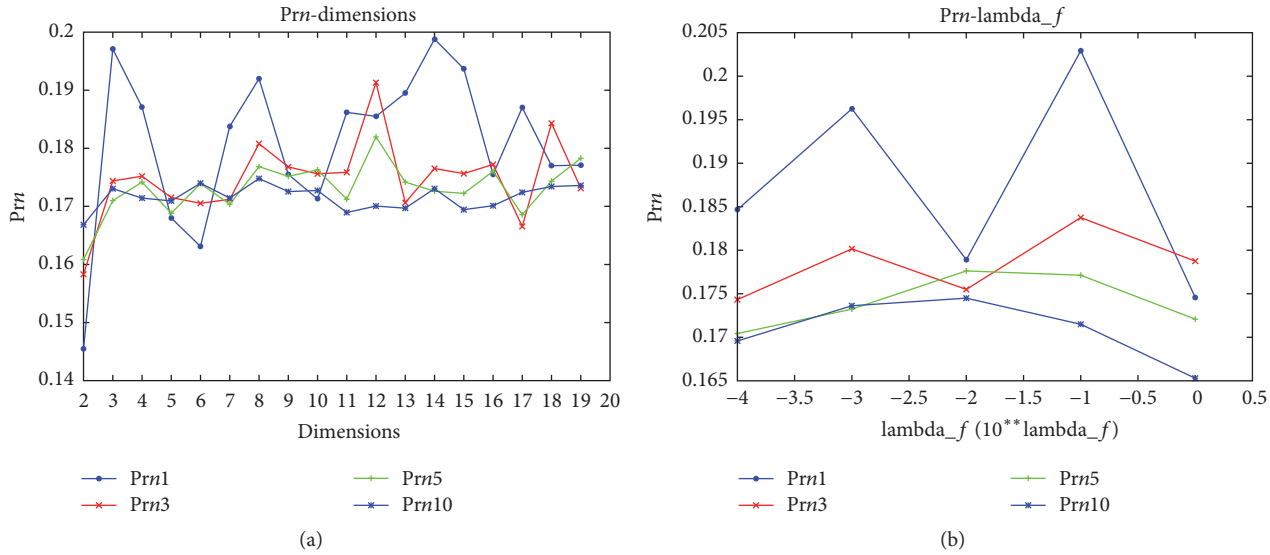


Figure 6: Impact of different values of different parameters in the SocialMF model on performance of user interest prediction.

According to Figure 5, we can get a set of parameters $d = 12$ and $\lambda_U = \lambda_V = 0.01$, which can make the models perform better on the top-1, top-3, top-5, and top-10 accuracy rate.

(2) SocialMF Model. The SocialMF model has four parameters, $\lambda_U, \lambda_V, \lambda_F, d$, in this paper. $\lambda_U, \lambda_V, \lambda_F$ are the regularization term coefficients in the loss function. In order to reduce the computational complexity, we set $\lambda_U = \lambda_V = 0.01$ which we set in the first experiment, and we set $\lambda_F = 0.001$ before setting parameters. d is the dimension of the latent features which is generally less than the rank of the original matrix. The control variable method is used to set the parameters by fixing other values and changing one. Then we can draw a graph to get the impact of each parameter on top-n accuracy. The top-n accuracy varies with the parameter d as shown in Figure 6(a) and with the parameter λ_F as shown in Figure 6(b).

Based on Figure 6, we can get a set of parameters $d = 12$, $\lambda_U = \lambda_V = 0.01$, and $\lambda_F = 0.1$ which can make the model have better performance on the top-1, top-3, top-5, and top-10 accuracy rate.

(3) TS-PMF Model. The TS-PMF model has six parameters, $\lambda_U, \lambda_V, \lambda_F, d, \theta$, and β. λ_U, λ_V, and λ_F are the regularization term coefficients in the loss function. In order to reduce the computational complexity, we set $\lambda_U = \lambda_V = 0.01$ which we set in the first experiment, and we set $\beta = 3$ and $\lambda = 0.001$ before setting parameters. d is the dimension of the latent features which is generally less than the rank of the original matrix. θ, β are the parameters in the forgotten function. The

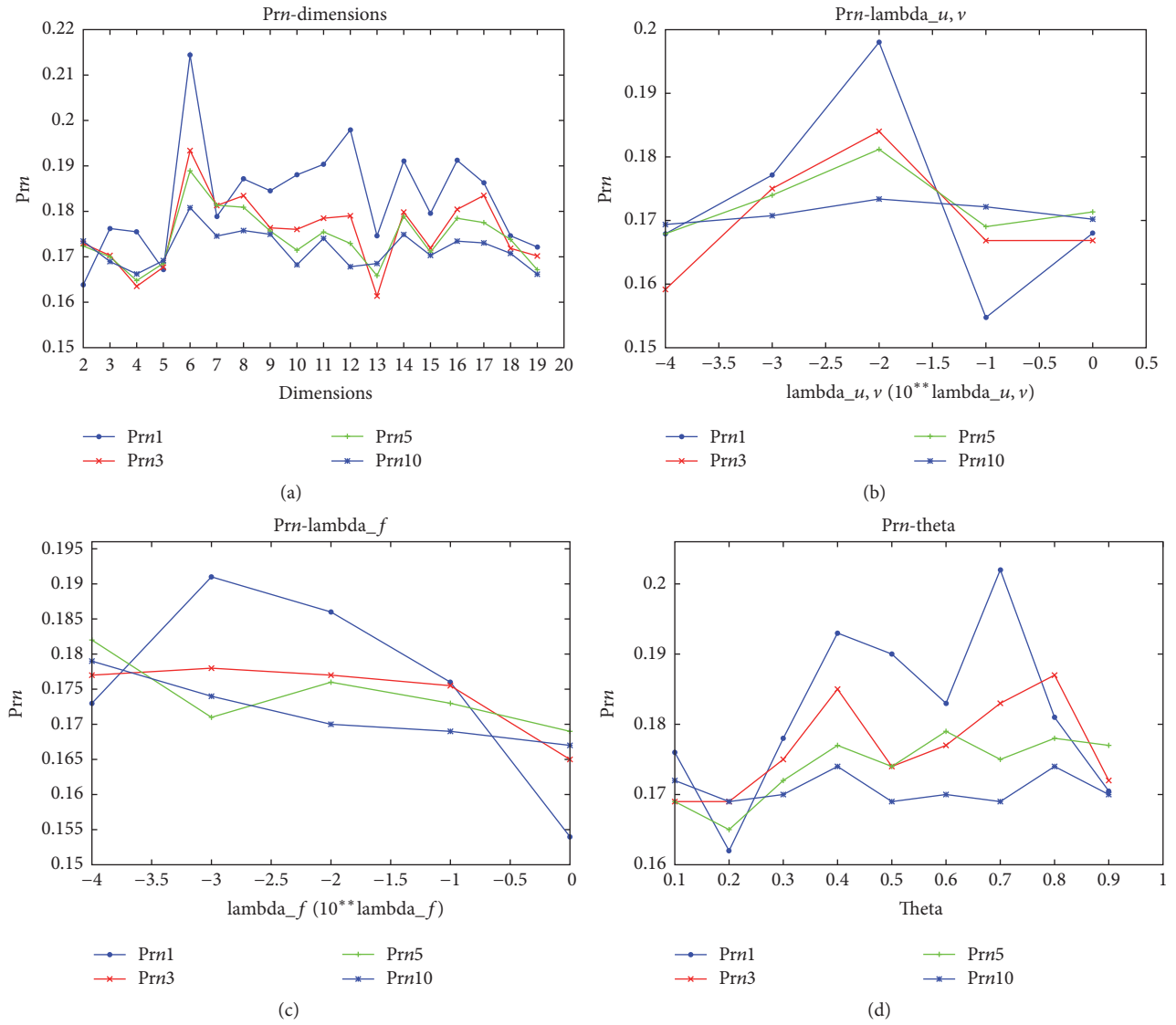


Figure 7: Impact of different values of different parameters in the TS-PMF model on performance of user interest prediction.

control variable method is used to set the parameters by fixing other values and changing one. Then we can draw a graph to get the impact of each parameter on top-n accuracy. The top-n accuracy varies with the parameters λ_U, λ_V, λ_F, d, and θ as shown in Figure 7.

From Figure 7, we can get a set of parameters $\lambda_U = \lambda_V = 0.01$, $\lambda_F = 0.001$, $d = 6$, and $\theta = 0.4$ which can make the model have better performance on the top-1, top-3, top-5, and top-10 accuracy rate.

(4) SHMF Model. The SHMF model has ten parameters, λ_{U_1}, λ_{V_1}, λ_{F_1}, λ_{U_2}, λ_{V_2}, λ_{F_2}, d, θ, β, and λ. In order to reduce the computational complexity, according to independence hypothesis, "the information of blogs posted by users and the information of blogs posted by others in user's social hub influence the user's interest in the future independently"; we can set $\lambda_{U_1} = \lambda_{V_1} = 0.01$ and $\lambda_{F_1} = 0.001$ in accordance with the third experiment. Then we set $\beta = 3$ and $\lambda_{U_2} = \lambda_{V_2}$, so we should actually consider the five parameters $\lambda_{U_2} = \lambda_{V_2}$, λ_{F_2}, d, θ, and λ, in which λ_{U_2}, λ_{V_2}, and λ_{F_2} are the regularization term coefficients in the loss function. d is the dimension of the latent features which is generally less than the rank of the original matrix. θ, β are the parameters in the forgotten function. λ indicates how important the user's social hub information is to the user's interest. We set $\lambda = 0$ if only user's personal posting behavior is considered and the SHMF model degrades to TS-PMF model at this time, and we set $\lambda = 1$ if only user's social hub information is considered. The control variable method is used to set the parameters by fixing other values and changing one. Then we can draw a graph to get the impact of each parameter on top-n accuracy. The top-n accuracy varies with the parameters λ_{U_2}, λ_{V_2}, λ_{F_2}, d, θ, and λ as shown in Figure 8.

According to Figure 8, we can get a set of parameters $\lambda_{U_1} = \lambda_{V_1} = 0.1$, $\lambda_{F_1} = 0.0001$, $d = 6$, $\theta = 0.3$, and $\lambda = 0.5$

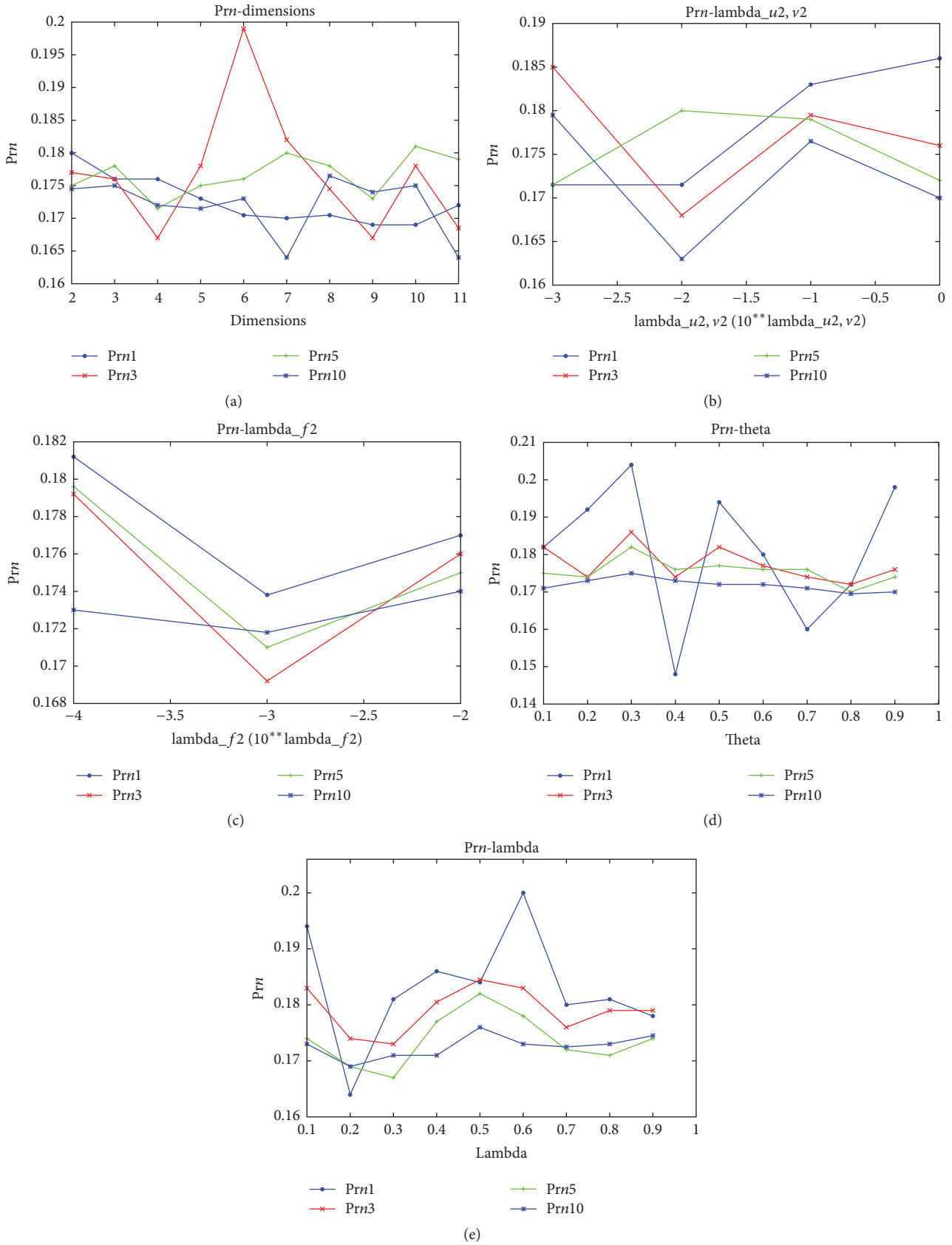


Figure 8: Impact of different values of different parameters in the SHMF model on performance of user interest prediction.

Table 1: Precision of SHMF.

	Pre_avg
PMF	17.35%
SocialMF	17.37%
TS-PMF	17.91%
SHMF	18.67%

Table 2: Performance of SHMF.

	Run-time (s)
PMF	698.618
SocialMF	1227.088
TS-PMF	1721.555
SHMF	2080.513

which can make the model have better performance on the top-1, top-3, top-5, and top-10 accuracy rate.

7.3. Experimental Results and Analysis

(1) Comparison of Accuracy. After adjusting the parameters of the five models, it is necessary to compare the strengths and weaknesses of the different models. As a result of the fact that the selection of different top-n accuracy will lead to different results, in order to consider comprehensively, this paper takes top-1, top-3, top-5, and top-10 accuracy of the arithmetic mean as the average accuracy, as shown in the following equation:

$$\mathrm{Pre}_{\mathrm{avg}} = \frac{\mathrm{Pre}_1 + \mathrm{Pre}_3 + \mathrm{Pre}_5 + \mathrm{Pre}_{10}}{4}. \tag{20}$$

By adjusting the model parameters of five experiments, the average accuracy of the five models under most parameters is shown in Table 1.

It can be seen from Table 1 that the algorithm SHMF proposed in this paper improves the average accuracy by over 1.3% compared to algorithm PMF and algorithm SocialMF and the average accuracy of the algorithm SHMF is 0.76% higher than the algorithm TS-PMF.

(2) Executive Efficiency Analysis. On the efficiency of implementation, based on the best parameters, set the number of iterations to 100 times and record the run-time, as shown in Table 2.

It is found from Table 2 that the running time of the algorithm SHMF is the longest, which is nearly three times the running time of the algorithm PMF. This is because, with the calculation of the complexity of the increase, the run-time of the algorithm SHMF has increased.

(3) Result Analysis. Through the comparison of four groups of experiments, we can see the difference and relation of PMF-based algorithm in microblog users' interest prediction. In the first comparative experiment, we use the most basic probability matrix factorization algorithm and got the average accuracy of 17.35%. In the second comparative experiment, the social trust relationship is added based on the probability matrix factorization algorithm. However, the average accuracy is almost the same as that obtained by the basic probability matrix factorization algorithm. This is mainly due to the fact that, in constructing dataset, we take the users whose posts are in a certain range and then determine their social trust relationships according to the statistical characteristics instead of using all or as many social trust relationships as possible for a user in order to consider both the similarity of behavior and the mutual influence among users. Therefore, this kind of method leads to sparsity of social trust matrix, so the impact is relatively small. Since we do not only focus on the correlation between users, we use this approach to implement the experiment. Compared with the previous two experiments, the average accuracy of the third comparative experiment is higher than that of the previous two experiments, and it is proven that the fact that this method based on the short-term interest of users is changing along time is rational. In the last experiment, the algorithm SHMF proposed in this paper will improve the average accuracy rate of nearly one percentage point, indicating that the user's social hub information does affect the user's interest in microblog and verifying the effectiveness of the algorithm at the same time.

8. Conclusions and Future Work

Based on the work of the prediction of microblog users' interest, this paper analyzes the information of microblog users' social hub and puts forward the SHMF model, which greatly improves the top-n accuracy and average accuracy. This will lay the foundation for the follow-up research work. At the same time, we can solve the cold-start problem of predicting interests of the users who do not often post blogs by analyzing the information of their social hub. This method could have a broad application space in social platform recommendation. However, there are still some defects in the implementation efficiency. When the amount of data is particularly large, the running time is too long, which needs to be improved in the future work.

For the future work of microblog users' interest prediction, further research on the expression of interest should be carried out to achieve more accurate representation, which determines the upper limit of interest prediction. In the prediction algorithm, we should add more techniques, such as Bayesian analysis, to solve the multiparameter problem by analyzing the relationship between the parameters and the actual meaning.

Notations

R_{1t}: The user-topic matrix in time t
R_{2t}: The user's social hub-topic matrix in time t
F_1: The user-user matrix
F_2: The hub-hub matrix
U_{1t}^T: The users' latent feature space in time t
V_{1t}^T: The topics' latent feature space in time t
U_{2t}^T: The users' latent feature space in social hub in time t

V_{2t}^T: The topics' latent feature space in social hub in time t
U_t^T: The final users' latent feature space in time t
V_t^T: The final topics' latent feature space in time t
$M_{U_{1t}}$: The mean matrix of U_{1t} with spherical Gaussian priors in time t
$M_{U_{2t}}$: The mean matrix of U_{2t} with spherical Gaussian priors in time t
$M_{V_{1t}}$: The mean matrix of V_{1t} with spherical Gaussian priors in time t
$M_{V_{2t}}$: The mean matrix of V_{2t} with spherical Gaussian priors in time t
θ: A weight that indicates how important the whole previous time points are to the current one
β: The kernel parameter
d: The dimension of latent feature space
λ: A weight that indicates how important the user's social hub information is to the user's interest
λ_{U_1}: The impact of the users' latent feature vectors on users' interests
λ_{U_2}: The impact of the social hubs' latent feature vectors on users' interests
λ_{V_1}: The impact of the topics of the blogs posted by users on users' interests
λ_{V_2}: The impact of the topics of the blogs posted by others in users' social hub on users' interests
λ_{F_1}: The impact of the users' relationships on users' interests
λ_{F_2}: The impact of the social hubs' relationships on users' interests.

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China (31371340) and the National Key Technologies Research and Development Program of China (no. 2016YFB0502604).

References

[1] X. Tang, C. C. Yang, and M. Zhang, "Who will be participating next? Predicting the participation of dark web community," in *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics 2012, ISI-KDD 2012*, Beijing, china, August 2012.

[2] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization. In NIPS 2008, volume 20".

[3] H. Ma, H. Yang, and M. R. Lyu, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pp. 931–940, Napa Valley, Calif, USA, October 2008.

[4] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the 4th ACM Recommender Systems Conference (RecSys '10)*, pp. 135–142, Barcelona, Spain, September 2010.

[5] H. Y. Zhang, L. W. Wang, and Y. X. Chen, "Research progress of probabilistic graphical models: a survey," *Journal of Software. Ruanjian Xuebao*, vol. 24, no. 11, pp. 2476–2497, 2013.

[6] G.-F. Sun, L. Wu, Q. Liu, C. Zhu, and E.-H. Chen, "Recommendations based on collaborative filtering by exploiting sequential behaviors," *Ruan Jian Xue Bao/Journal of Software*, vol. 24, no. 11, pp. 2721–2733, 2013.

[7] H. Bao, Q. Li, S. S. Liao, S. Song, and H. Gao, "A new temporal and social PMF-based method to predict users' interests in micro-blogging," *Decision Support Systems*, vol. 55, no. 3, pp. 698–709, 2013.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[9] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the the seventh ACM SIGKDD international conference*, pp. 57–66, August 2001.

[10] R. R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," in *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[11] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2001.

[12] X. W. Meng, S. D. Liu, Y. J. Zhang, and X. Hu, "Research on social recommender systems," *Journal of Software. Ruanjian Xuebao*, vol. 26, no. 6, pp. 1356–1372, 2015.

[13] L. Guo, J. Ma, Z.-M. Chen, and H.-R. Jiang, "Incorporating item relations for social recommendation," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 37, no. 1, pp. 219–228, 2014.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 3956415, 7 pages
https://doi.org/10.1155/2017/3956415

Research Article

A Quick Negative Selection Algorithm for One-Class Classification in Big Data Era

Fangdong Zhu,[1] Wen Chen,[1,2] Hanli Yang,[3] Tao Li,[1,2] Tao Yang,[1] and Fan Zhang[1,4]

[1]*College of Computer Science, Sichuan University, Chengdu 610065, China*
[2]*College of Cybersecurity, Sichuan University, Chengdu 610065, China*
[3]*Chongqing University of Technology, Chongqing 400054, China*
[4]*Chengdu University of Information Technology, Chengdu 610225, China*

Correspondence should be addressed to Wen Chen; wenchen@scu.edu.cn and Hanli Yang; yhl@cqut.edu.cn

Received 2 February 2017; Accepted 3 May 2017; Published 12 June 2017

Academic Editor: Zonghua Zhang

Negative selection algorithm (NSA) is an important kind of the one-class classification model, but it is limited in the big data era due to its low efficiency. In this paper, we propose a new NSA based on Voronoi diagrams: VorNSA. The scheme of the detector generation process is changed from the traditional "Random-Discard" model to the "Computing-Designated" model by VorNSA. Furthermore, we present an immune detection process of VorNSA under Map/Reduce framework (VorNSA/MR) to further reduce the time consumption on massive data in the testing stage. Theoretical analyses show that the time complexity of VorNSA decreases from the exponential level to the logarithmic level. Experiments are performed to compare the proposed technique with other NSAs and one-class classifiers. The results show that the time cost of the VorNSA is averagely decreased by 87.5% compared with traditional NSAs in UCI skin dataset.

1. Introduction

NSA was proposed by Forrest et al. in 1994 [1], which generates immune detectors based on the "Random-Discard" model. Initially, massive immature detectors are randomly generated, and then the ones covering the self-areas are discarded. González et al. presented the real-valued negative selection algorithm (RNSA) in 2003 [2], in which the detectors and antigens are studied in the real-value space. Ji and Dasgupta proposed V-Detector algorithm [3, 4]. It turns the fixed-length detectors in RNSA into the variable-sized detectors to enlarge the detection areas. In 2015, Cui et al. developed BIORV-NSA [5]. In their work, the self-radius can be variable and the detectors, which are recognized by other mature detectors, are replaced by new ones to eliminate the "detection holds."

In big data era, the low efficiency of NSA becomes an important challenge, which largely limits its applications. In this paper, we design a new NSA based on Voronoi diagrams, named VorNSA. In the VorNSA, a restrained Voronoi diagram is constructed based on the whole training set in the first step. Then, two types of detectors are generated in the specific location of the Voronoi diagram separately. In order to accelerate the test stage of NSA, in particular for large scale dataset, a new testing strategy VorNSA/MR (VorNSA with Map-Reduce) is proposed. Unlike the testing stage of classic NSAs, data are divided into small groups and calculated to generate the labels separately in Map stage. Then the final labels can be obtained after merging and sorting in the Reduce stage.

The contributions of this work can be summarized as follows. (1) Based on Voronoi diagrams, the optimal position of detectors is calculated directly rather than in a stochastic way. Therefore, the time consumption wasted on excessive invalid detectors is avoided. (2) In the Map/Reduce framework, data are partitioned into several small parts by VorNSA/MR and can be processed in parallel to enhance the self/non-self-discrimination efficiency.

The rest of the paper is organized as follows. In Section 2, we describe the definitions of VorNSA. The original contribution of the paper is presented in Section 3. Experimental results on synthetic datasets and real-world datasets are

shown and discussed in Section 4. Conclusions appear in Section 5.

2. Basic Definition of VorNSA

VorNSA is designed based on Voronoi, which is derived from computation geometry to search the nearest neighbors, and it has been widely utilized in the fields of life sciences [6], material sciences [7], and mobile navigation [8]. The basic definitions are listed as follows.

Definition 1 (site). Site is a set of n distinct points in the feature space. In VorNSA, all the training samples are defined as site points: $S = \{S_1, S_2, \ldots, S_n\}$.

Definition 2 (Voronoi diagram). Vor(S) divides the feature space into n unoverlapped cells based on the given site set S, and each cell $\nu(S_i)$ only contains one site S_i in S, such that any point q in $\nu(S_i)$ satisfies $\text{dist}(q, S_i) < \text{dist}(q, S_j)$ $\forall S_j \in S$, $j \neq i$, and dist( ) can be any distance metrics.

Definition 3 (cell). All the cells construct a mathematic partition of the feature space, and the cell corresponding to site S_i is denoted by $\nu(S_i)$.

Definition 4 (largest empty circle). The largest circle with center p, which does not contain any site in S, is denoted by $C_S(p)$.

Theorem 5. *A point p is a vertex of* Vor(S) *iff $C_S(p)$ contains at least three sites on its boundary [9].*

Definition 6 (I-detector). $\langle c, r\rangle$, where c is the detector position in the feature space, and r is the detector radius, satisfies that c corresponds to one vertex of the Voronoi diagram.

Theorem 7. *Given p is the center of an I-detector, there are at least three sites located on the boundary of $C_S(p)$, and these sites are the nearest neighbors of each other.*

Proof. According to Definitions 2 and 6, it can be inferred that the center of the I-detector p is an intersection of three or more cells. Suppose that p is intersected by three cells $\nu(S_i)$, $\nu(S_j)$, $\nu(S_k)$, while the sites of these cells are S_i, S_j, S_k. According to Definition 4 and Theorem 5, there is a largest empty circle $C_S(p)$ that does not contain any site of S, and S_i, S_j, S_k are located on its boundary. So S_i, S_j, and S_k are the nearest sites of p among the site sets S. □

Theorem 8. *The bisector between sites S_i and S_j defines an edge of* Vor(S) *iff there is a point q on the bisector such that $C_S(q)$ contains both S_i and S_j on its boundary with no other site [9].*

Definition 9 (II-detector). $\langle c, r\rangle$, where c is the detector position in the feature space, and r is the detector radius, satisfies that c corresponds to the junction of the edges of Vor(S) and the unit hypercube $[0, 1]^d$.

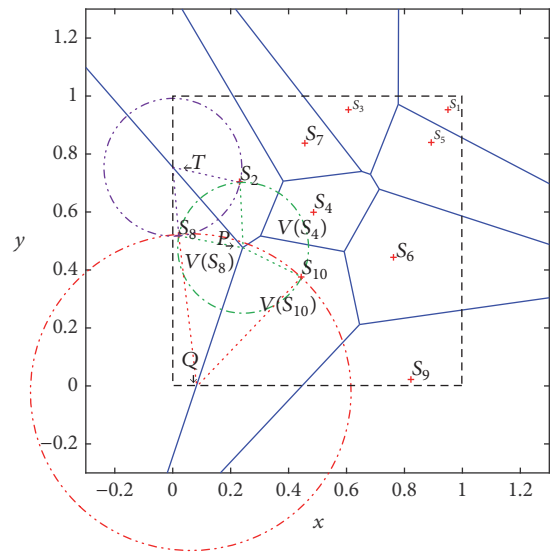


Figure 1: The red cross is the site. The blue line is the Voronoi diagram. The circle is the largest empty circle.

Theorem 10. *Given q is the center of II-detector, there are two sites located on the boundary of $C_S(p)$, and these sites are the nearest neighbors of each other.*

Proof. According to Definitions 2 and 9, it can be inferred that the center of II-detector q is an intersection of two cells. Suppose that q is intersected by two cells $\nu(S_i)$, $\nu(S_j)$, while the sites of these cells are S_i, S_j. According to Definition 4 and Theorem 8, there is a largest empty circle $C_S(q)$ that does not contain any site of S, and S_i, S_j are located on its boundary. So S_i and S_j are the nearest sites of q among the site sets S. □

As an example in Figure 1, there are 10 sites S_1–S_{10} in set S, and the space is divided into 10 cells $\nu(S_1)$–$\nu(S_{10})$ by the Voronoi diagram Vor(S). The green circle is $C_S(P)$, and three sites (S_2, S_8, S_{10}) are located on its boundary. The red circle is $C_S(Q)$, and two sites (S_8, S_{10}) are located on the boundary. The purple circle is $C_S(T)$, and two sites (S_2, S_8) are located on the boundary. P is the center of I-detector, while Q and T are the centers of II-detector.

3. The Details of VorNSA

3.1. The Detector Generation Process of VorNSA

3.1.1. Space Partition Stage. First of all, all the training data are normalized to $[0, 1]^d$ feature space, where d is the data dimension. The normalized training set is denoted by S. Secondly, a bounded Voronoi diagram Vor(S) is constructed based on S, to divide the unit feature space $[0, 1]^d$ into n cells, where $n = |S|$. Finally, the set $VS = \{\langle \text{Vet}(S_i), S_i\rangle \mid i = 1, \ldots, n\}$, where $\text{Vet}(S_i)$, S_i are the vertex and site in a cell $\nu(S_i)$, can be constructed.

3.1.2. I-Detector Generation Stage. According to Definition 6 and Theorem 7, the center of I-detector p is designated by the

Input: Training set S, Self radius R_S, Minimum detector radius δ
Output: Detector set D
(1) normalize S into $[0, 1]^d$
(2) construct voronoi diagram Vor(S) by sites S
(3) get all cells $\nu(S_i)$ in Vor(S)
(4) construct $VS = \{\langle \text{Vet}(S_i), S_i\rangle \mid i = 1, \ldots, n\}$ by $\nu(S_i)$
(5) foreach $\langle \text{Vet}(S_i), S_i\rangle$ in VS
(6) if Vet(S_i) has three or more same values in VS
(7) then $VS1 = VS1 \cup \langle \text{Vet}(S_i), S_i\rangle$
(8) foreach $\langle \text{Vet}(S_j), S_j\rangle$ in $VS1$
(9) compute the detector radius R_P using Eq. (1)
(10) if $R_P > \delta$ then $D_{\text{I}} = D_{\text{I}} \cup \langle \text{Vet}(S_j), R_P\rangle$
(11) foreach $\langle \text{Vet}(S_i), S_i\rangle$ in VS
(12) if Vet(S_i) has two same values in VS
(13) then $VS2 = VS2 \cup \langle \text{Vet}(S_i), S_i\rangle$
(14) foreach $\langle \text{Vet}(S_k), S_k\rangle$ in $VS2$
(15) compute the detector radius R_Q using Eq. (2)
(16) if $R_Q > \delta$ then $D_{\text{II}} = D_{\text{II}} \cup \langle Q_i, R_{Qi}\rangle$
(17) return $D = D_{\text{I}} \cup D_{\text{II}}$

Algorithm 1: VorNSA (S, R_S, δ).

intersection of three or more cells, and the sites located in the cells are the nearest neighbors of each other. So a new set $VS_1 = \{\langle \text{Vet}(S_j), S_j\rangle | j = 1 \cdots\}$, where Vet($S_j$) is the position of I-detector and S_j is the nearest sites, can be obtained by $\text{Vet}(S_j) = \{x \mid x = \text{Vet}(S_p) \cap \text{Vet}(S_q) \cap \text{Vet}(S_t),\ p \neq q \neq t\}$, where Vet($S_p$), Vet($S_q$), and Vet($S_t$) are the vertex sets of cell. Then, generating a mature detector is just through self-tolerating with S_j. According to the principle of self-tolerance, the radius of I-detector can be calculated with

$$R_P = \text{dist}\left(\text{Vet}\left(S_j\right), S_j\right) - R_S, \tag{1}$$

where R_P is the radius of I-detector, Vet(S_j) is the center of I-detector, S_j is the nearest sites, and R_S is the radius of self-antigens.

Furthermore, a threshold δ of detector radius is introduced in case of overfitting: If the detector radius R_P is less than δ, the detector will be discarded. Otherwise, it will mature.

3.1.3. II-Detector Generation Stage. The main difference between the I-detector and the II-detector is the location of detector centers. According to Definition 9 and Theorem 10, the position of II-detector q is located on the junction of two cells and the unit hypercube. The sites in the two cells are the nearest neighbors of each other. So a new set $VS_2 = \{\langle \text{Vet}(S_k), S_k\rangle \mid k = 1 \cdots\}$, where Vet($S_k$) is the position of II-detector and S_k is the nearest sites, can be obtained by $\text{Vet}(S_k) = \{x \mid x = \text{Vet}(S_p) \cap \text{Vet}(S_q),\ p \neq q\}$, where Vet($S_p$) and Vet($S_q$) are the vertex sets of cell. Similarly, the radius of II-detector can be computed by (2), and a threshold δ of detector radius is introduced in case of overfitting.

$$R_Q = \text{dist}\left(\text{Vet}\left(S_k\right), S_k\right) - R_S, \tag{2}$$

where R_Q is the radius of II-detector, Vet(S_k) is the position of II-detector and S_k is the nearest sites, and R_S is the radius of self-antigens.

Details of the VorNSA can be found in Algorithm 1.

3.2. The Immune Detection Process of VorNSA under Map/Reduce Framework. In the testing stage of traditional NSAs, each piece of data has to be compared with all the detectors to label its classification. This strategy is too time-consuming to be applied in big data era due to its low efficiency. In order to enhance the efficiency in testing stage, an immune detection process of VorNSA under Map/Reduce framework (VorNSA/MR) is proposed. Map/Reduce is a parallel computation framework, which splits the sample set into a group of small datasets and handles them on many cluster nodes simultaneously.

Details of VorNSA/MR (Figure 2) are mainly divided into two parts: Map stage and Reduce stage. First of all, the testing datasets are split into n parts by VorNSA/MR. In the Map stage, each cluster node selects a part of split data to compute the distance with matured detectors. If any distance is less than the detection radius, the testing sample is labeled with the non-self-antigens; otherwise it is labeled with the self-antigens. Then cluster nodes put results to the intermediate value. The Reducer receives the intermediate values, sorts them, and merges them into the final results.

The implements of Map and Reduce stage can be found in Algorithms 2 and 3.

3.3. Theoretical Analysis

Theorem 11. *The time complexity of VorNSA is* $(N_S \log N_S + N_S^{\lceil d/2\rceil} + |D|)$*, where* N_S *is the size of training dataset,* d *is the dimension of training dataset, and* $|D|$ *is the size of detectors.*

```
Input: Detector set D, Split data T
Output: Intermediate Value IV
(1) foreach T_i in T
(2)         foreach D_k in D
(3)                 Compute the Euclidean distance dist(T_i, D_k) between T_i and D_k
(4)                 if D_k.r  < dist(T_i, D_k)
(5)                       T_i is Noself Antigen, T_i.Label = 0
(6)                       go to line (2)
(7)                 T_i is Self Antigen, T_i.Label = 1
(8) IV.Value = ⟨T.no, T.Label⟩
(9) return IV
```

Algorithm 2: Mapper (D, T).

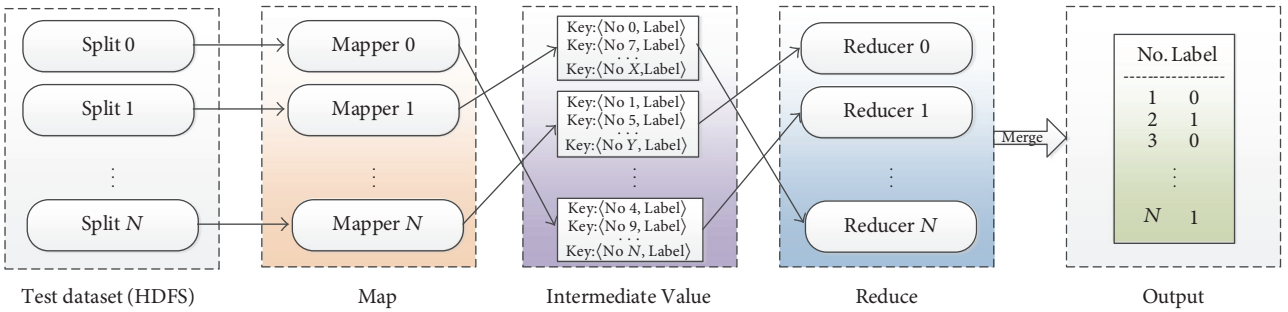


Figure 2: The details of VorNSA/MR.

```
Input: Intermediate Value IV
Output: Final Value FV
(1) While IV.next ~= END
(2)         add IV.Value to FV.Value
(3) Sort FV.Value by no
(4) return FV
```

Algorithm 3: Reducer (IV).

Proof. Since VorNSA is divided into three stages, we could analyze the time complexity separately.

The main work in space partition stage is to build a Voronoi diagram, so we borrow the analysis from Voronoi diagrams to estimate the time complexity. The literatures [9–12] prove that a Voronoi diagram with n sites can be computed in $O(n \log n + n^{\lceil d/2 \rceil})$ optimal time under d-dimension space. Therefore, the time complexity can be denoted by $O(N_S \log N_S + N_S{}^{\lceil d/2 \rceil})$, where N_S is the size of training set, and d is the dimension of training set.

In the second and third stage, the main work is to compute the distance between detectors and sites. Though several detectors are discarded by the threshold δ, the quantity is very small compared with the whole size, so we use the size of detectors $|D|$ instead. According to (1) and (2), we can infer that the time complexity is $O(|D|)$ in the two stages.

Combining the abovementioned, the time complexity of VorNSA is $O(N_S \log N_S + N_S{}^{\lceil d/2 \rceil} + |D|)$. □

Table 1: The complexity of NSAs.

Algorithm	Time complexity
NNSA [1]	$O\left(-\frac{\ln P_f * N_S}{P_m (1 - P_m)^{N_S}}\right)$
RNSA [2]	$O\left(\frac{\lvert D\rvert * N_S}{(1 - P_m)^{N_S}}\right)$
V-Detector [4]	$O\left(\frac{\lvert D\rvert * N_S}{(1 - P_m)^{N_S}}\right)$
VorNSA	$O\left(N_S \log N_S + N_S{}^{\lceil d/2 \rceil} + \lvert D\rvert\right)$

The time complexity of traditional NSAs is shown in Table 1, where P_m is the match probability between detectors and antigens, P_f is the failure rate, N_S is the size of self-set, $|D|$ is the size of detectors, and d is the data dimension. As shown in Table 1, the time complexity of VorNSA is in logarithmic level with N_S, which is much less than the traditional exponential level compared with NNSA [1], RNSA [2], and V-Detector [4].

4. Experiments and Discussion

In the experiments, we use two evaluation criteria of performance: DR (Detection Rate) and FAR (False Alarm Rate) which is reported in varied literature [2, 3, 13], and they are defined as

$$\begin{aligned} \mathrm{DR} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \\ \mathrm{FAR} &= \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \end{aligned} \tag{3}$$

Table 2: The detail of 4 SDS.

Dataset	Records number	Self-antigens	Non-self-antigens
Cross	10,000	5,531	4,469
Ring	10,000	3,710	6,290
Pentagram	10,000	2,850	7,150
Triangle	10,000	1,476	8,524

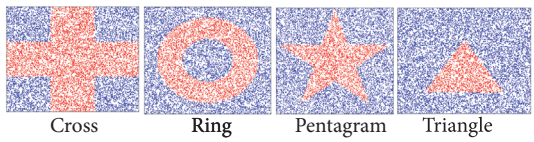


Figure 3: The distribution of 4 SDS.

where TP and FN are the counts of true positive and false negative of non-self-antigens, respectively, and TN and FP represent the number of true negative and false positive of self-antigens, respectively.

4.1. Experiments on Synthetic Dataset (SDS). In order to determine the performance of VorNSA among different datasets, 4 SDS proposed by the intelligence security laboratory of Memphis University are introduced in this section. The records of original datasets [3] are 1000, respectively. We expand the number of pieces of data to 10,000 to simulate the environment of big data better. The distributions of datasets are depicted as Figure 3 in which self-antigens are represented by red dots and non-self-antigens are shown by blue points. The details of datasets are listed in Table 2. Additionally, experiment parameters are set as follows: the self-radius is 0.04, self-antigens are randomly obtained from 50 to 1000, and the minimum radius of detectors is 0.005. Each experiment is repeated 25 times independently.

As Figure 4 shows, the trends of experiment results on 4 SDS are approximately the same. It indicates that VorNSA could achieve a high degree of applicability on different datasets. In Figure 4(a), it can be observed that the DR decreases from 95% to 80% with the increment of self-antigens. Besides, in Figure 4(b), the FAR drops from 60% to zeros. The reasons of this phenomenon can be explained as follows: when less self-antigens are trained, some self-antigens cannot be covered by the scope of self. So these self-antigens are identified as non-self-antigens in VorNSA. Due to its strong ability in detecting, the DR and FAR are both high. With the increase of the training numbers, all self-antigens will be covered. Furthermore, the non-self-antigens are covered and identified as self-antigens, in particular those located in the edge of self-set. Therefore, the DR decreases slightly while FAR sharply drops to zeros.

Figure 4(c) shows the quantity of detectors generated by VorNSA is not increasing remarkably with the growth of train set but maintains a relatively stable range. It is implied that VorNSA can effectively control the expansion of detectors. According to Definition 2, with the increment of training samples, the space will be partitioned into smaller cells. We introduce the minimum detector radius δ. Thus, the inefficient tiny detectors are discarded.

In Figure 4(d), it can be noted that the time consumption of VorNSA on different datasets is similar, and time cost rises slowly even with enormous self-antigens. It suggests that the performance of VorNSA is less affected by the distribution of dataset, because the optimal position of detectors is calculated directly rather than in a stochastic way.

To sum up, we can see that VorNSA can generate fewer but more effective detectors. Besides, the less self-antigens are trained, the higher FAR will be. With the number of self-antigens increasing, the FAR is decreased significantly. Increasing the training set will lead to a rise of the time consumption, and the DR will be slightly decreased. Hence, a smaller self-set will be a smart choose in VorNSA.

4.2. Experiments on Skin Segmentation Dataset. In this section, VorNSA is tested by a group of comparison experiments. The compared algorithms include the classic NSAs (RNSA, V-Detector), a newly proposed NSA (BIORV-NSA) in 2015. To study the different methods, we introduce a classic statistics algorithm for one-class classification: OC-SVM [14], which is implemented by LibSVM [15]. All algorithms run in a computer deployed with Intel Pentium E6600@3.06 G, while the implement of VorNSA refers to an open source toolbox of computational geometry, called MPT 3.0 [16].

The Skin Segmentation dataset is a UCI dataset. It is collected by randomly sampling B, G, and R values of skin texture, which derives from FERET database and PAL database. Total sample size is 245,057 in which 50,859 records are the skin samples and 194,198 records are non-skin ones.

In this experiment, 50 skin samples are randomly obtained as self-antigens. Meanwhile, to verify the performances of VorNSA and VorNSA/MR in large scale dataset, we use all 245,057 records in the datasets. The experiments are preformed 20 times independently, and the evaluation criteria include DR, FAR, detector number (DN), data training time (DT), and data testing time (DTT). The parameters of simulation are set as follows: the OC-SVM uses the RBF kernel functions, and nu is 0.5 and gamma is 0.33. The self-radius of RNSA, V-Detector, and VorNSA are set as the same value (0.1). The maximum number of detectors is 3000 in RNSA, and detector radius is 0.1. The estimated coverage and the maximum self-coverage are 99%. The maximum number of detectors is 1000 in BIORV-NSA, and the self-set edge inhibition parameter is 0.8 and the detector self-inhibition parameter is 1.2. The minimum radius of detectors is 0.005 in VorNSA and VorNSA/MR. The results of experiments are shown in Table 3.

From Table 3, it can be seen that the FAR of OC-SVM is 51.2%, reaching an unacceptable level. As OC-SVM implemented in a different platform, the time consumption is not counted in this paper. The DR of VorNSA (99.2%) is closed to the BIORV-NSA (99.42%), and better than the classic NSAs. Besides, the FAR of VorNSA (1.48%) is lower than BIORV-NSA (3.29%). It indicates that the detectors generated by VorNSA are more applicable than BIORV-NSA and more effective than classic NSAs.

Table 3: Results in skin segmentation.

Algorithm	DR (%)		FAR (%)		DN		DT (s)		DTT (s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
OC-SVM	99.09	0.7	51.20	6.67	—	—	—	—	—	—
RNSA	98.42	0.63	**0.66**	1.48	3000.00	0	8.68	0.13	7501.59	400.49
V-Detector	99.05	0.27	1.31	1.22	469.85	174.66	32.12	23.86	948.55	325.50
BIORV-NSA	99.42	0.34	3.29	2.72	1000.00	0	20.00	0.11	1919.83	59.46
VorNSA	99.20	0.16	1.48	1.49	**172.25**	11.06	1.91	0.77	671.15	89.36
VorNSA/MR*	**99.43**	0.24	1.56	1.37	176.90	11.96	**1.79**	0.07	**426.70**	31.97

*The VorNSA/MR is deployed at 2 nodes: one is Intel Pentium E6600@3.06 G (2 Core); the other is Inter Core i5-2450M@2.5 G (2 Core).

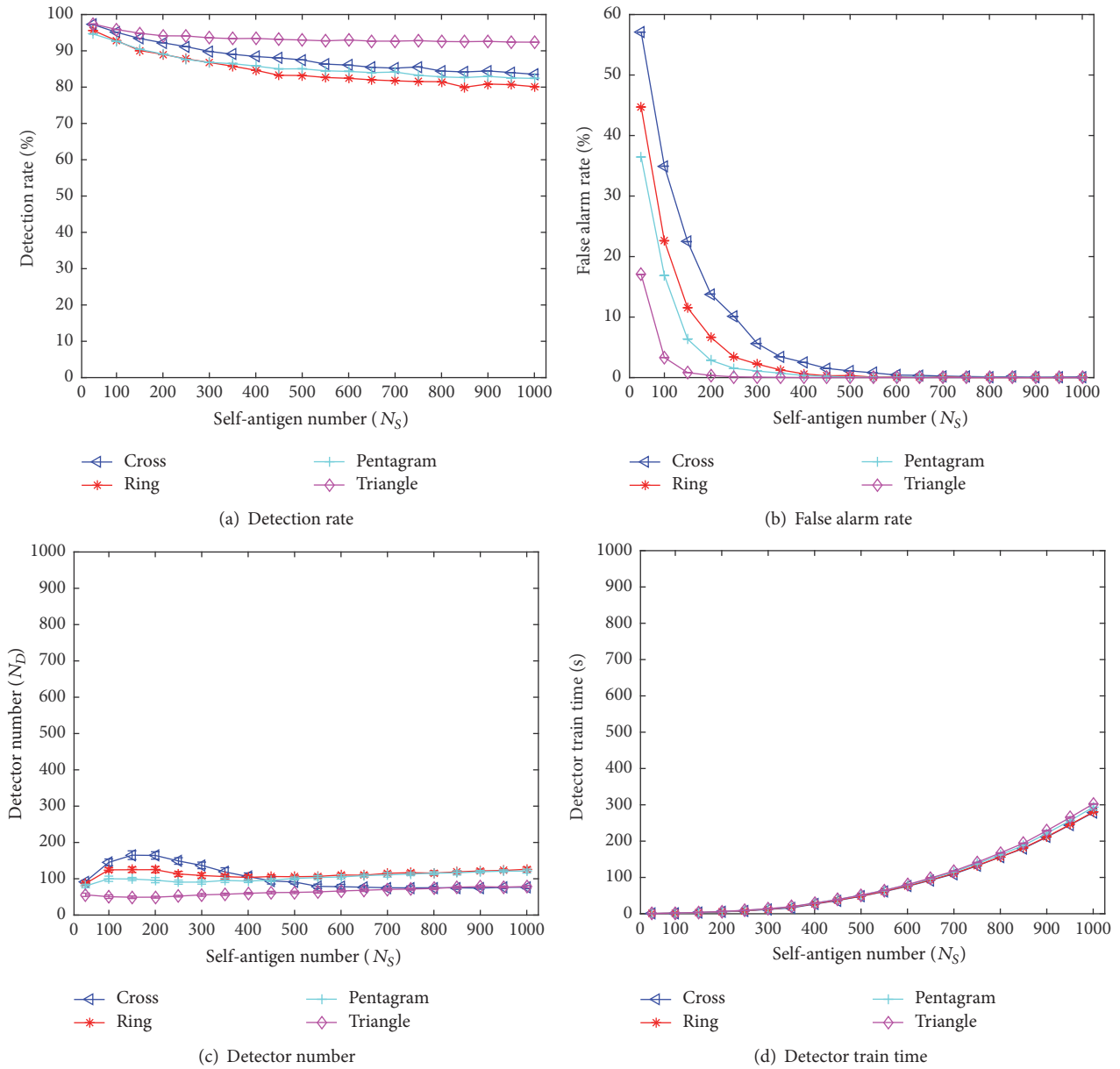


Figure 4: Results with different training samples.

Moreover, the DN, DT, and DTT of VorNSA are significantly lower than other NSAs, especially when it integrates the Map-Reduce Testing Framework. For example, the average number of detectors generated by VorNSA is 172.25, lower 63.3% by V-Detector and 82.8% by BIORV-NSA. The average training time of VorNSA is 1.91, lower 78% by RNSA, 94.1% by V-Detector, and 90.5% by BIORV-NSA. So the efficiency of VorNSA is averagely decreased by 87.5% compared with traditional NSAs. The testing time of VorNSA/MR is 426.7, lower 36.4% by VorNSA, 55% by V-Detector, 77.8% by BIORV-NSA, and 94.3% by RNSA.

The main reasons of above results can be explained as follows. In traditional NSAs, a large number of immature detectors are randomly generated without any optimal way and must self-tolerate with all self-antigens to decide whether they are matured or not. As a result, much time has been wasted. The scheme of detector generation of VorNSA is quite different with other NSAs. The optimal position of detectors is directly calculated. Thus, the time consumption on discarding many randomly generated but inappropriate detectors is avoided.

5. Conclusions

In this paper, we propose a new one-class classification algorithm based on Voronoi diagrams (VorNSA) and an immune detection process of VorNSA under Map/Reduce framework (VorNSA/MR) to cope with the challenge of big data. VorNSA alters the generative mechanism of detector from the "Random-Discard" model to the "Computing-Designated" model. VorNSA/MR can divide the sample set into several small parts and can be processed in parallel. Theoretical analyses show that the time complexity of VorNSA decreases from the exponential level to the logarithmic level. Experiments results show that the time consumption of VorNSA is significantly declined.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant nos. 2016YFB0800605 and 2016YFB0800604) and Natural Science Foundation of China (Grant nos. 61402308 and 61572334).

References

[1] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the IEEE Symposium on Research in Security and Privacy, (SP '94)*, pp. 202–212, IEEE Computer Society, Oakland, May 1994.

[2] F. González, D. Dasgupta, and L. F. Niño, "A randomized real-valued negative selection algorithm," in *In Proceedings of the 2nd International Conference on Artificial Immune Systems*, vol. 2787, pp. 261–272, 2003.

[3] Z. Ji and D. Dasgupta, "Real-valued negative selection algorithm with variable-sized detectors," in *Genetic and Evolutionary Computation Conference*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 287–298, Springer, Berlin, Heidelberg, 2004.

[4] Z. Ji and D. Dasgupta, "V-detector: an efficient negative selection algorithm with 'probably adequate' detector coverage," *Information Sciences*, vol. 179, no. 10, pp. 1390–1406, 2009.

[5] L. Cui, D. Pi, and C. Chen, "BIORV-NSA: Bidirectional inhibition optimization r-variable negative selection algorithm and its application," *Applied Soft Computing Journal*, vol. 32, pp. 544–552, 2015.

[6] D. Sanchez-Gutierrez, M. Tozluoglu, J. D. Barry, A. Pascual, Y. Mao, and L. M. Escudero, "Fundamental physical cellular constraints drive self-organization of tissues," *EMBO Journal*, vol. 35, no. 1, pp. 77–88, 2016.

[7] H. W. Sheng, W. Luo, F. Alamgir, J. Bai, and E. Ma, "Atomic packing and short-to-medium-range order in metallic glasses," *Nature*, vol. 439, pp. 419–425, 2006.

[8] G. Zhao, K. Xuan, W. Rahayu et al., "Voronoi-based continuous nearest neighbor search in mobile navigation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2247–2257, 2011.

[9] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, Springer, 2008, https://www.amazon.com/Computational-Geometry-Applications-Mark-Berg/dp/3540779736.

[10] B. Chazelle, "An optimal convex hull algorithm and new results on cuttings," in *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pp. 29–38, October 1991.

[11] K. L. Clarkson and P. W. Shor, "Applications of random sampling in computational geometry, II," *Discrete & Computational Geometry*, vol. 4, no. 1, pp. 387–421, 1989.

[12] R. Seidel, "Small-dimensional linear programming and convex hulls made easy," *Discrete & Computational Geometry*, vol. 6, no. 1, pp. 423–434, 1991.

[13] W. Chen, T. Li, X. Liu, and B. Zhang, "A negative selection algorithm based on hierarchical clustering of self set," *Science China Information Sciences*, vol. 56, no. 8, pp. 1–13, 2013.

[14] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class SVM for learning in image retrieval," in *Proceedings of IEEE International Conference on Image Processing (ICIP) 2001*, pp. 34–37, grc, October 2001.

[15] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, article 27, 2011.

[16] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proceedings of the 12th European Control Conference, (ECC '13)*, pp. 502–510, Zurich, Switzerland, July 2013.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4541975, 12 pages
https://doi.org/10.1155/2017/4541975

Research Article

Economic Levers for Mitigating Interest Flooding Attack in Named Data Networking

Licheng Wang,[1] Yun Pan,[2] Mianxiong Dong,[3] Yafang Yu,[4] and Kun Wang[2]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*School of Computer Sciences and Technology, Communication University of China, Beijing 100024, China*
[3]*Department of Information and Electronic Engineering, Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan*
[4]*Anyang Normal University, Anyang, Henan 455002, China*

Correspondence should be addressed to Licheng Wang; wanglc2012@126.com

Received 14 February 2017; Accepted 18 April 2017; Published 7 June 2017

Academic Editor: Zonghua Zhang

As a kind of unwelcome, unavoidable, and malicious behavior, distributed denial of service (DDoS) is an ongoing issue in today's Internet as well as in some newly conceived future Internet architectures. Recently, a first step was made towards assessing DDoS attacks in Named Data Networking (NDN)—one of the promising Internet architectures in the upcoming big data era. Among them, interest flooding attack (IFA) becomes one of the main serious problems. Enlightened by the extensive study on the possibility of mitigating DDoS in today's Internet by employing micropayments, in this paper we address the possibility of introducing economic levers, say, dynamic pricing mechanism, and so forth, for regulating IFA in NDN.

1. Introduction

Today's Internet is a unique and unprecedented global success story [1]. It is built based on TCP/IP architecture and assumes that users and ends are trustable and intelligent, and the main task of the Internet is to provide best effort service of packet forwarding. This idea caters to the original requirements on mutually connecting hosts and sharing distributed resources. However, with the increasing and flourishing of the models of computations and applications, the way people access and utilize the Internet has changed dramatically, and today's Internet is reaching the limits of their senescence [1]. To keep pace with changes and move the Internet into the future, several projects have been initiated to design potential next-generation Internet architectures [1].

In 1999, Adjie-Winoto et al. [2] proposed the concept of "Content-Centric." Afterwards, more researchers have been paying efforts on this direction, and the idea of Information Centric Networking (ICN) is widely accepted, now. With ICN, each piece of information has a unique name as its identity, by which users can request consuming desired information, while the network needs only to manage the flowing and cache these pieces of information according users requests and information's names. In other words, with ICN, users need only to know what he/she wants, instead of where the information is located. Names themselves carry less information about routing than IP addresses used in today's Internet. Recently, big ICN research projects are mainly distributed in Europe and America, such as Date-Oriented Transfer (DOT) architecture [3], Data-Oriented Network Architecture (DONA) [4], Routing on Flat Labels (ROFL) [5], Internet Indirect Infrastructure (or i3 for short) [6], Publish-Subscribe Internet Routing Paradigm (PSIRP) [7], Content-Centric Networking (CCN) [8–10], 4WARD [11], and TRIAD [1]. Among them, Content-Centric Networking (CCN) due to Jacobson et al. [8–10] is currently a comparatively mature architecture. In particular, CCN*x* [10] is an open-source suite that enables more researchers to put forward their improvements as well as CCN-based new applications [12]. In recent years, the project Named Data Networking (NDN) [13],

with thoroughly integrating the idea of ICN/CCN, made remarkable progress, including a series of typical applications [14, 15], as well as NS-3 friendly simulation tools for further development [16]. In particular, in the upcoming big data era, NDN will inevitably become one of the promising Internet architectures due to is data-centric features.

In order to avoid past pitfalls, security experts insist that we should treat security and privacy as fundamental requirements, and in particular resilience to denial of service (DoS) and distributed denial of service (DDoS) attacks become a major issue and deserve full attention during conceiving next-generation Internet architectures [1]. Recently, Gasti et al. [1] made a first step towards assessing DDoS attacks in NDN. On one hand, many kinds of DoS/DDoS attacks that have heavy impact on today's Internet are successfully bypassed due to subtleties and exactitude of designing of NDN. In particular, the pulling model and the receiver-driven mechanism used in NDN make most DoS/DDoS attacks becoming aimless (i.e., it is difficult to find victims), and the mechanism of reverse path content delivering makes most DoS/DDoS attacks reflect to themselves. But as the proverb goes, "every coin has its two sides," NDN has not uprooted DoS/DDoS attacks. Gasti et al. also conceived two kinds of new DoS/DDoS attacks that intentionally utilize the features of NDN: interest flooding attack (IFA) and content/cache poisonous attack (CPA). Shortly afterwards, Atanasyev et al. [17] showed that NDN's inherent property of flow balancing provides the basis for effectively mitigating IFA.

However, as far as we know, little attention is paid to mitigating IFA in NDN by employing micropayment systems. But we know that in fighting against DoS/DDoS attacks on today's Internet, micropayments have been extensively studied during the past two decades [18]. The idea of micropayments in fighting against DoS/DDoS attacks focuses on incurring heavy penalties such as "virtual money" (say, CPU cycles, memory/disk, bandwidth, etc.) to the DoS/DDoS attackers. Therefore, in this paper, we try to probe the possibility of using economic levers, such as micropayments and different pricing functions, to deal with the interest flooding attacks in NDN. Our discussion mainly includes three parts: a prototype of economic model for NDN, evaluation on knowing types of micropayments in NDN, and assessing the possible utilities of knowing pricing functions in NDN. In addition, we also address the possibility of charging content producers and relate this issue to the area of digital right management (DRM).

The rest of content is organized as follows: in Section 2, we give a brief introduction on NDN and IFA; in Section 3, our main contribution, a prototype of economic model for NDN, is proposed; finally, the concluding remarks are found in Section 5.

2. Reviewing NDN and Interest Flooding Attacks

As a typical instance of the broader ICN/CCN approach to networking, NDN aims to evolve it into an architectural framework for the future Internet [1]. NDN eliminates host-based addressing and explicitly names content and thus transforms content into a first-class entity [17]. Based on this abstraction there is no explicit notion of "hosts" in NDN, although their existence is assumed. Instead, interest and content are the only two types of packets in NDN, and each NDN router maintains three major data structures [1]:

(i) Pending Interest Table (PIT), a table containing currently unsatisfied interests and corresponding incoming interfaces

(ii) Forwarding Interest Base (FIB), a table containing name prefixes and corresponding outgoing interfaces

(iii) Content Store (CS), a buffer used for content caching and retrieval

Based on these components, communication in NDN takes the *pull* model: A consumer requests content by sending an interest packet; if an entity (a router or a host) can fetch from his CS a matched content object (i.e., named data packet), the corresponding data packet will be returned to the consumer by following the reverse path of the interest request [17]. These features make NDN a receiver-driven, data-centric communication protocol [17] and thus automatically bypass several long-standing DoS/DDoS attacks, such as direct flooding and reflector attacks through source address spoofing [17].

However, in 2012, Gasti et al. conceived the so-called interest flooding attacks (IFA) that utilize the features of NDN: the adversary, with controlling of a large set of zombies, invokes a large number of interest requests that are distributed closely in space, aiming to overflow PITs in routers, preventing them from handling legitimate interests, and/or to swamp the specific content producer(s) [1]. Gasti et al. further identified three types of IFA based on the whether the requested content exists and how the content produced [1]:

(I) Existing and static

(II) Dynamically generated

(III) Nonexistent

As for IFA with type (I), the impact on NDN routers is limited since in-network content caching mechanism will automatically block subsequent same/similar interest requests not to propagate to the producer(s). As for IFA with type (II), the impact on NDN routers varies with respect to their distance from the targeted content producer(s): the closer the router to the producer(s), the greater the effect on its PIT [1]. IFA with type (III) cannot incur significant overhead for targeted content producer(s), but unsatisfied interest requests will propagate to other NDN nodes and the corresponding PIT entries will be occupied with longest time—until they eventually expire [1].

3. A Prototype of Economic Model for NDN

It is a common belief that a resource may be abused if its users incur little or no cost [19]. Thus, it is reasonable to introduce payments or in general micropayments into NDN for fighting IFA. In fact, the idea of requiring the user to commit its resources before requesting services was described early by

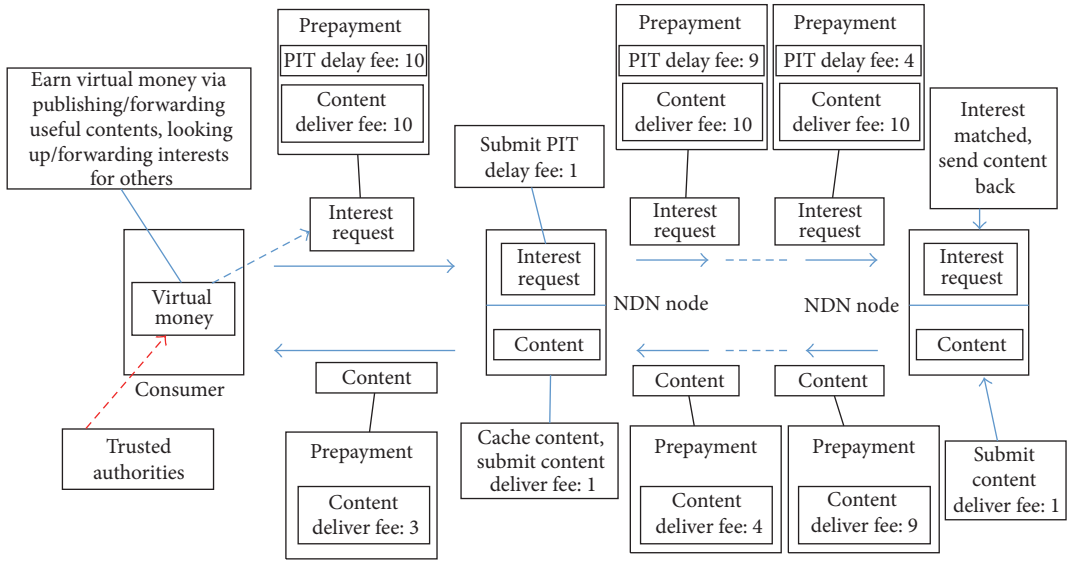


Figure 1: The proposed prototype of economic model for NDN.

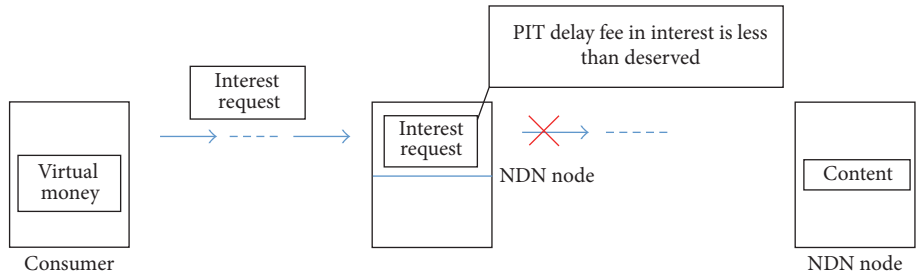


Figure 2: Dropping interest request due to lack of PIT delay fee.

Dwork and Naor [20, 21]. As early as about 10 years ago, Mankins et al. [18] once introduced dynamic resource pricing models for mitigating distributed denial of service attacks. But their models were conceived under the scenarios with typical TCP/IP architectures, and thus some aspects need to be updated for NDN architecture accordingly.

3.1. Business Logics. For mitigating IFA in NDN, the proposed prototype of economic model is featured by the following business logics:

(1) Suppose that there are trusted authorities in NDN, and they do not only play the role of central banks for issuing virtual money (VM) and related strategies, but also conduct related tasks like auditing, accounting, and so on (as analogy of reality, one might prefer to assign the duties of auditing and accounting to other trusted authorities, instead of banks; but this has no essential effects on our prototype).

(2) Suppose each user or NDN node possesses certain amount of VM at the beginning, and he/she can earn more VM via publishing/forwarding useful contents, looking up/forwarding interests for others.

(3) Each user is required to submit his/her prepayment (PP), as long as prompting an interest request. This prepayment includes two parts: PIT delay fee (PDF) and content delivering fee (CDF).

(4) Upon receiving an interest request from some downstreaming node that might be an end consumer or a NDN router the NDN node i looks up his/her local cache for interest matching: if failing, then make allowance for PIT delay fee, denoted by pdf_i, and then forward the interest request to all/part of upstreaming nodes; if matched, then make allowance for content delivering fee, denoted by cdf_i, and then transfer the content to the requester via a reverse path along the interest request; and every NDN node j in this path will also make allowance for content delivering fee cdf_j and meanwhile keep the content in his/her local cache (see Figure 1).

(5) Each NDN node can stop and discard interests forwarding if the left prepayment carried by the request package is less than his/her charging on PIT delay fee (see Figure 2). Similarly, each NDN node can stop contents forwarding (i.e., the red crossing

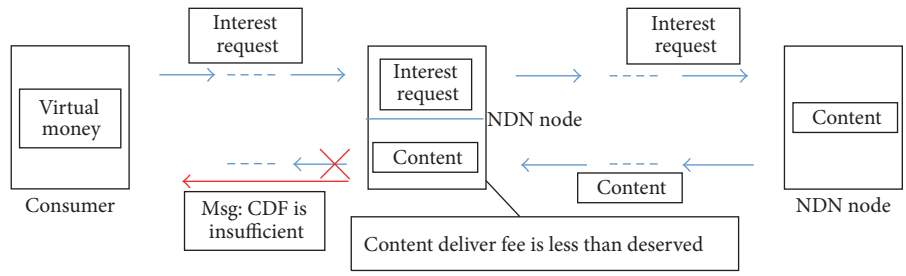


Figure 3: Stopping content delivering due to lack of prepayments.

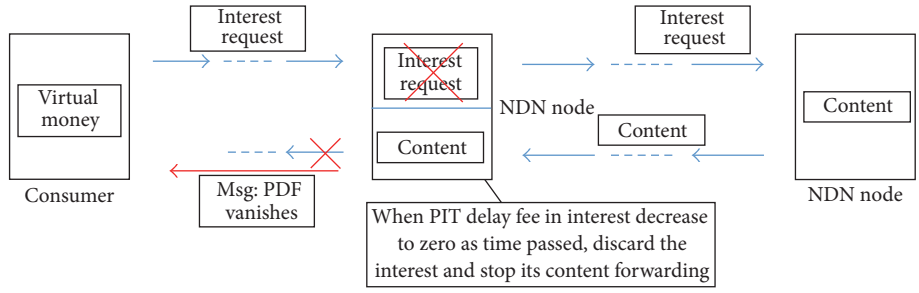


Figure 4: Stopping content delivering due to PIT delay fee vanishes.

symbol in Figure 3) if the left prepayment is less than his/her charging on content delivering fee. This is reasonable since the forward node, as well as the downstream nodes, has no obligation to delivering packages without earnings. However, this node need not immediately discard this kind of undelivered content. Instead, he/she can choose to cache this content for short period and meanwhile send a short message "CDF is insufficient" to the requester via a reverse path along the interest request. This kind of short message can be regarded as special "contents" packets and the related content delivering fee is set to zero.

(6) PIT delay fee vanishes with its delay time in PIT table. In other words, as for some item in a PIT table, its PIT delay fee pdf_i will decrease along time elapse, and the NDN node will discard this PIT item if this $\text{pdf}_i \leq 0$. When this occurs, the NDN node can also send another short message "PDF vanishes" to the requester via a reverse path along the interest request. Similarly, this kind of short message can also be regarded as special "contents" packets and the related content delivering fee is set to zero. Meanwhile, the two red crosses in Figure 4 indicate that the related forwarding processes are also cancelled. This is reasonable considering that some nodes might become unreachable after he/she sends requests. In this case, it is useful to space the PIT buffers for accommodate newly coming requests.

(7) All involved economic behavior should be auditable and accountable. Enforcing each NDN node to sign his/her actions or responses related to VM provides a good support for achieving postauditing and accounting. Auditing and accounting should be executed by some trusted authorities periodically.

Remark 1. Compared to the original NDN architecture, the processes of delivering the above two kinds of short messages are newly introduced. Based on the following observations, we think these new additions are compatible with the original NDN architecture and useful for improvement the performance.

(i) If a NDN router node directly discards related PIT entries in local PIT table but without sending the short message "CDF is insufficient" or the short message "PDF vanishes" then we return to the original NDN settings.

(ii) Upon receiving either of these two special messages, an end user can choose to resend the same interest request with additional prepayments. Then, the interested contents might be fetched quickly in the midway.

(iii) Since these two short messages are transferred along the reverse path of interest requests, the downstreaming NDN nodes can take actions correspondingly:

(a) If the corresponding PIT entry still stays in local PIT table, then the NDN node can forward the incoming short messages downwards and then discard this PIT entry.

(b) Otherwise, if the corresponding PIT entry has already been discarded from local PIT table, then the NDN node no longer need forward the incoming short messages downwards, since before this occurs, it might have sent the short

message "PDF vanishes" along the reverse of the path of interest requests. Recursively, the related end users have the chance to receive at least one short message and this is sufficient for prompting him/her to resend the same interest request with additional payments.

Remark 2. Someone might argue whether the business logic depicted in Figure 3 is reasonable. Seemingly, it is unfair for the consumer because no service has been provided in this case. Someone is even afraid of the fact that based on this business logic a DoS attack can be mounted by sending interest requests with calculated insufficient CDF. However, we insist that the business logic depicted in Figure 3 is reasonable:

(i) Firstly, it is unfair for NDN routing nodes if in this case the consumer is not charged. Anyway, the involved NDN routing nodes have already done searching on related interests and even transferring contents during the network, although the contents have not reached the consumer. That is, we must pay NDN routing nodes. Without charging consumer, who pays that?

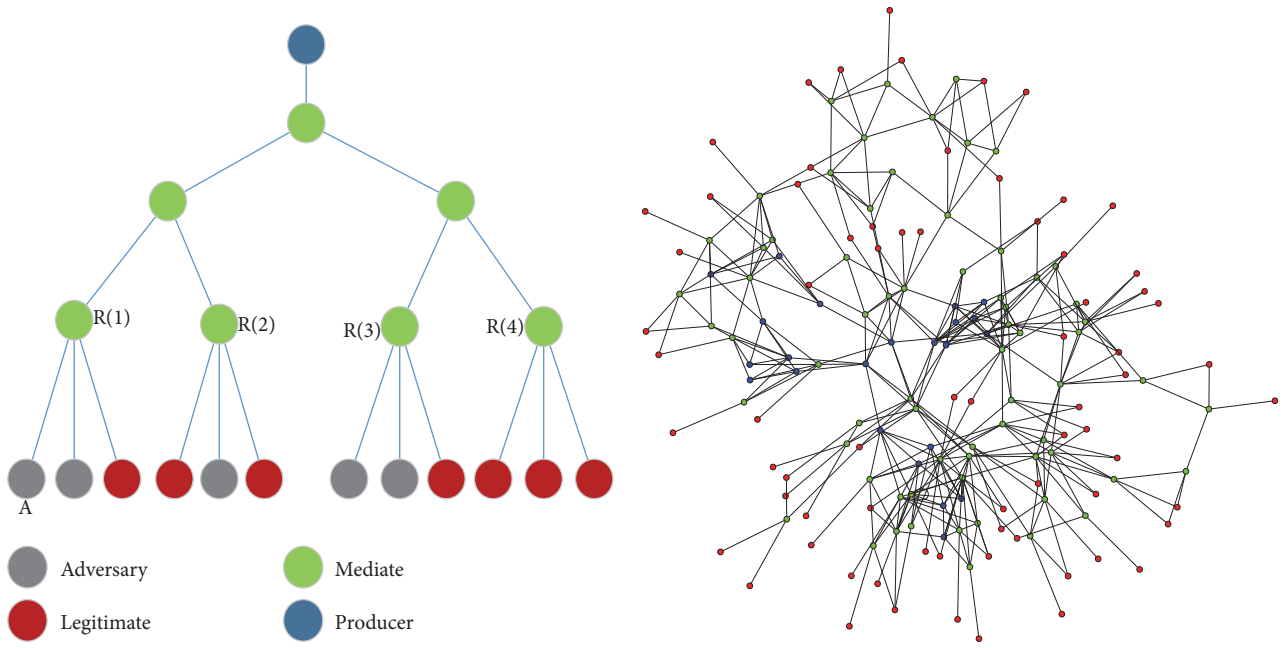
(ii) Secondly, even though the requested contents have not reached the consumer, the consumer obtains a useful message: CDF is insufficient. This message tell two facts to the consumer: (a) the interest request has been matched and (b) the requested content has already been stored in the halfway—this is just the core feature of NDN. That is, the consumer can launch the same interest request and then get the content from the halfway.

(iii) Thirdly, suppose one node, denoted by A, tries to mount a DoS attack by sending interest requests with calculated insufficient CDF. That means the prepayment of A should be large enough for routing NDN nodes find the matched contents; otherwise, the case in Figure 2, instead of the case in Figure 3, occurs. Now, suppose that the content is dropped in the halfway due to lack of CDF. Then, when A launches the same interest request again, also with insufficient CDF, now the request interest must be matched during the halfway. Again and again, the matched contents will come to A closer and closer. That is, the effects of this kind of DoS attack towards the whole network become less and less. Finally, when the content has merely one hop to A, this kind of DoS attack becomes useless.

3.2. Types of Micropayments. As addressed in [18], micropayments can provide a useful side benefit by providing a uniform means of resource accounting, pricing, and arbitration. But micropayments mechanisms must not impose an undue performance penalty. That is, the performance should be, in the absence of an attack, nearly comparable to a system that does not use the payment mechanisms [18]. There have been a number of digital payment and micropayment schemes to support digital exchanges [18, 22]. According to the description of the above prototype, we need fungible (or transferable) digital payment schemes. Among them, check or credit card-like schemes require some type of online verification of payment—a server connects online much with a bank and verifies the creditworthiness of the requester [18]. Apparently, this strategy is not suited for NDN since the server might become easily a bottle neck; cash-like schemes do not require online verification but require significant computation or memory usage overhead for validation [18] and thus may not be compatible with NDN-oriented applications; scrip-based system (such as Compaq's Millicent [23]) is featured in that the verification can be performed locally with very low latency and thus it is friendly to NDN-oriented applications. Note that today's popular digital cash BitCoin [24] might not be suited for NDN-oriented applications considering that it becomes more and more difficult to obtain a "coin"—this suggests that the mechanism of BitCoin does not provide a steady supply of currency with the flourishing of the applications in future. However, moderately hard, memory-bound functions suggested by Abadi et al. [19] might be useful. In particular, this kind of functions is evaluated at about the same speed on most popular systems like severs, laptops, PDAs, and so forth [19]. Recently, Shen et al. [21] suggested using retraffic strategy for fighting against DDoS in TCP/IP architecture. However, this method does not only rely on middle-software that is fixed in front of the server, but also request the client to send more traffic (i.e., retraffic) for a single request. After that, Khanna et al. [25] also proposed using bandwidth as currency. That is, in order to get service, the clients are encouraged to spend more bandwidth by either sending repeated requests or sending dummy bytes on a separate channel to enable a bandwidth auction [25]. However, as for NDN architecture, we state as a fact two obstacles for deploying these two methods: firstly, interest request in NDN is forwarded by NDN router nodes and the upstreaming nodes need not recognize the end client, and thus requesting the interrouter nodes to spend more bandwidth is irrational; secondly, where to deploy the newly introduced middle-software is not only a cost problem, but also an challenge with respect to modifying NDN architecture. Therefore, we are inclined not to use these two methods in NDN. In brief, we summarize the potential NDN compatibilities of different kinds of micropayments in Table 1.

Table 1: Compatibility of micropayments in NDN.

Types	Features/requirements	Compatibilities
Check/credit card-like	Online verification	Poor
Cash-like	Heavy local verification	Poor
Scrip-based	*Light local verification*	*Good* √
BitCoin-like	Lack of supply	Poor
Memory-bound functions	*Roughly same speed over different platforms*	*Good* √
Retraffic or bandwidth as payment	Clients are encouraged to spend more bandwidth	Poor



(a) A tree-like topology

(b) A net-like topology

Figure 5: Topologies for simulation.

3.3. Pricing Functions. It is also another common sense that we should employ a dynamic pricing strategy for each service, instead of a fixed pricing function for all services [18]. However, detailed addressing of this issue goes out the scope of this paper. As the first step towards analyzing possibility of using economic levers in NDN, we would like to abstractly classify all services in NDN into two categories: interest looking up and content delivering. In other words, from the view of NDN router nodes, all interests/contents in the above prototype have no much difference from random numbers. Their duties are just to look up, to forward, and to cache them. After that, these NDN routers will obtain what they deserved (i.e., VM) according certain charging policies. Note that this kind of abstraction does not exclude the following two possibilities: (1) pricing function may be time-varying according to NDN routers' capabilities and other situations of the network, like congestion and so forth; (2) Each end user has their *own* utility function that determines how much he/she is willing to pay for an interest request, although after submitting his/her interest request, all related NDN router nodes will charge PIT delay fee (i.e., pdf) and content delivering fee (i.e., cdf) regardless of which kind of interests/contents is requested/delivered. In fact, in our micropayment system, we can adopt the following price model:

$$\text{Price} = \max\{0, -U(\text{utility}) + C(\text{opportunity cost})\}, \quad (1)$$

where both the utility function U and the opportunity cost (this indicates the potential cost of giving bandwidth to the coming request while not giving to others) function C can be established in an adaptive manner, according to the long term competition and balance between the requests and the responses of NDN network services.

In the scenario of mitigating TCP SYN flooding attacks, Mankins et al. tested four different pricing functions [18]:

(i) Constant function ($p = k$): the price p is set to constant k regardless of its level of consumption.

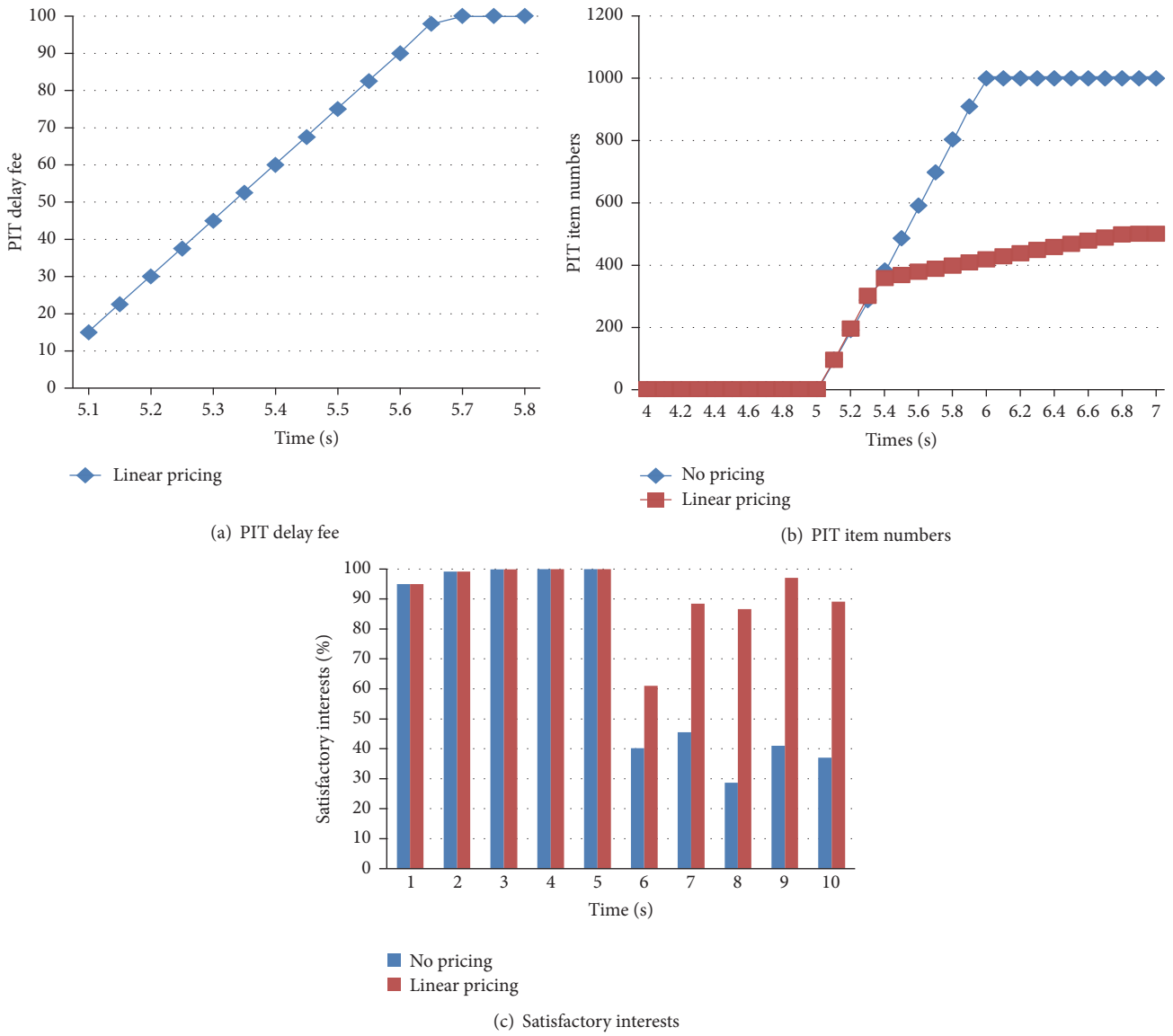
(ii) Linear function ($p = kc$): p is proportional to the value of a chosen market observable c such as the number of current connections.

(iii) Asymptotic function ($p = kB/(B - c)$): p is raised asymptotically to infinity as the market observable c approaches its limitation B.

(iv) Exponential function ($p = \alpha e^{\beta c}$): p is raised in the fastest manner with respect to the increasing value of the market observable c.

In fact, we can see that these pricing functions are reasonable in wide and universal scenarios and they are independent of concrete architectures. For example, the asymptotic pricing strategy is useful in safeguarding a resource with a hard limit in capacity, while the exponential pricing strategy is effective in controlling consumption of a critical resource [18]. The thing left is to consider how to use them, respectively, for mitigating interesting flooding attack in NDN.

(1) *Constant Pricing Function.* With the purpose of providing steady service, it seems that the simplest way is to use constant pricing strategy for forwarding incoming interest requests within the same time-window and with the same local connection degree. However, we think it is not suitable for our scenario: first, NDN architecture is topology-insensitive but constant pricing function should be, at least locally,



(a) PIT delay fee

(b) PIT item numbers

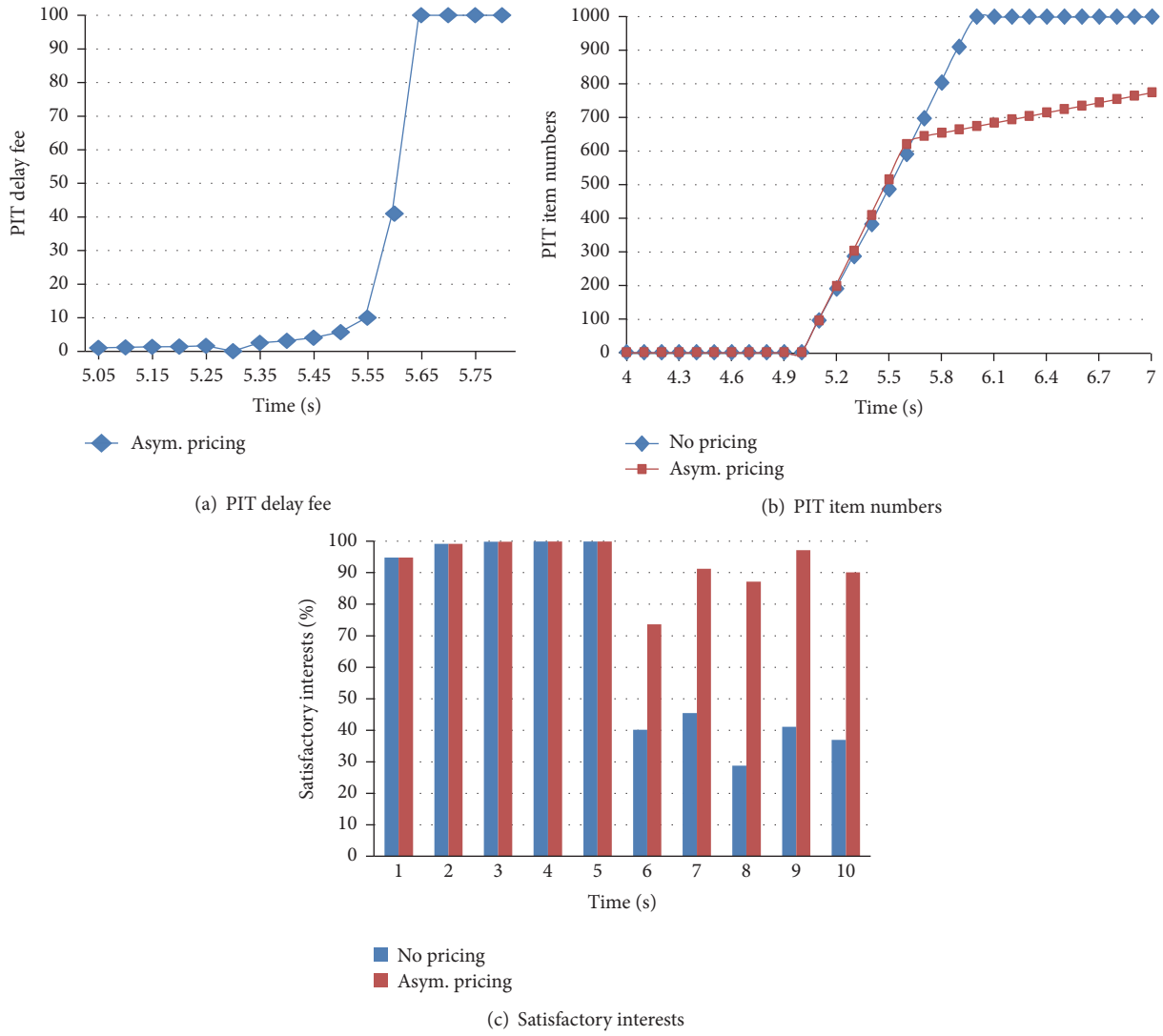
(c) Satisfactory interests

FIGURE 6: Simulation on linear pricing strategy ($k = 0.3$).

topology-aware; second, constant pricing function will charge IFA nodes with an unbiased mind, but our main motivation is to punish IFA nodes and the so-called unbiased mind towards malicious nodes will be *unfair for* legitimate nodes. Therefore, for mitigating IFA attacks, we will *not suggest using* constant pricing function.

(2) *Linear Pricing Function.* Since the concept of connection is not explicitly modeled in NDN architecture, we associate c in the related pricing functions to the number of interest requests coming from some ports. As a result, whenever a malicious node, denoted by $\mathscr{A}$, launches IFA attacks, the numbers of interest requests in PIT tables of $\mathscr{A}$'s upstreaming nodes increase linearly. This in turn induces linear increment of charging $\mathscr{A}$'s prepaid. When it is used out, the related interest request will be discarded. As for legitimate nodes, this kind of accumulation of interest request will not occur in PIT tables of the upstreaming nodes; thus the charge will be much small.

(3) *Asymptotic Pricing Function.* Here, c is also associated with the related pricing functions to the number of interest requests coming from some ports, while B is associated with the maximum number of interest requests that can be accepted by an upstreaming node. We will use asymptotic pricing function for *basically* charging PIT delay fee (i.e., pdf) (here, the term "basically" means the least charging without considering the further delay of PIT entries in PIT tables). That is, when the local PIT table becomes almost occupied, a NDN router node has to charge *hugely* for newly incoming interest requests. By using this



(a) PIT delay fee

(b) PIT item numbers

(c) Satisfactory interests

FIGURE 7: Simulation on asymptotical pricing strategy ($k = 1$, $B = 333$).

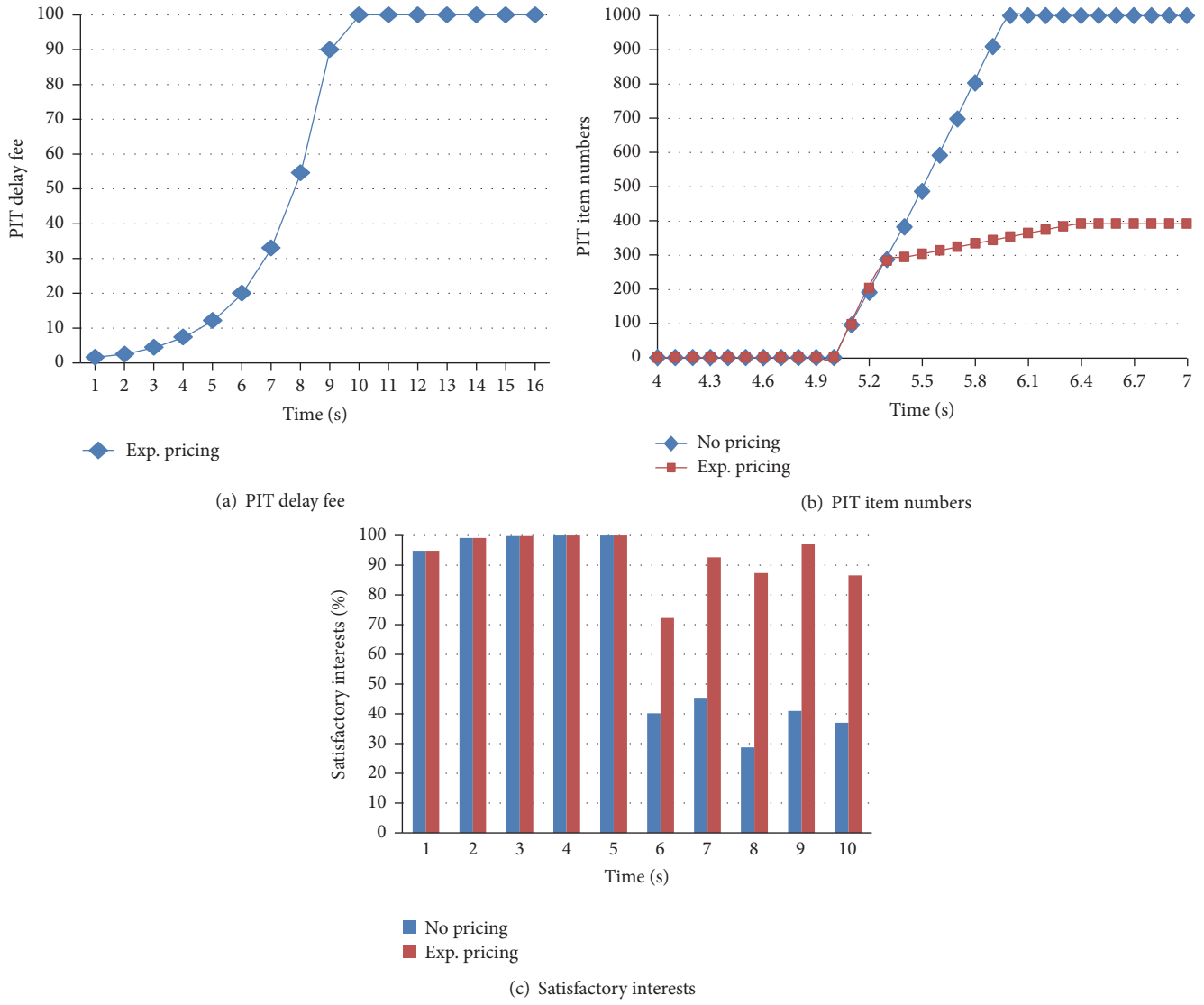
mechanism, downstreaming NDN nodes or end users are encouraged to submit/forward interest requests to those upstreaming nodes with more empty PIT entries. This is reasonable just like queue systems with multiple service windows in economic life.

(4) *Exponential Pricing Function.* The preserved PIT delay fee will be consumed according to exponential pricing function. This kind of charging can be viewed as *incremental* charging PIT delay fee and it will be an exponential function of delayed time in PIT table. This is rational since PIT entries are critical resource and thus cannot be occupied for long time by some "dead entries" (here, "dead entries" indicate those interest requests that cannot find matched contents).

To charge content delivering fee (i.e., cdf) in NDN, as well as in today's Internet, is a subtle problem. We know that bandwidth is also a critical resource. It seems that we should use exponential pricing function. However, this will encourage end users to split a single large request (say, "please download the whole book for me") into several small requests (say, "please download the ith chapter of the book for me") if they do not mind the delay of contents of the later chapters. This is unexpected since it runs in the opposite direction with respect to the "best effort" mechanism that is widely accepted in today's Internet and will continue to be useful in future Internet architectures, including NDN. Therefore, we suggest using asymptotic pricing function for charging content delivering fee. Partial reason for doing this is that within the same time-window and with the same local topology of network bandwidth has fixed limitation and from the view of NDN router node, local available bandwidth might be less critical than PIT entries.

In summary, the utilization of different pricing functions in NDN can be tabulated in Table 2.

3.4. Paying or Charging Content Producers? Seemingly, it is also reasonable to pay content producers, just like in economic life. However, since NDN architecture tries to play



(a) PIT delay fee

(b) PIT item numbers

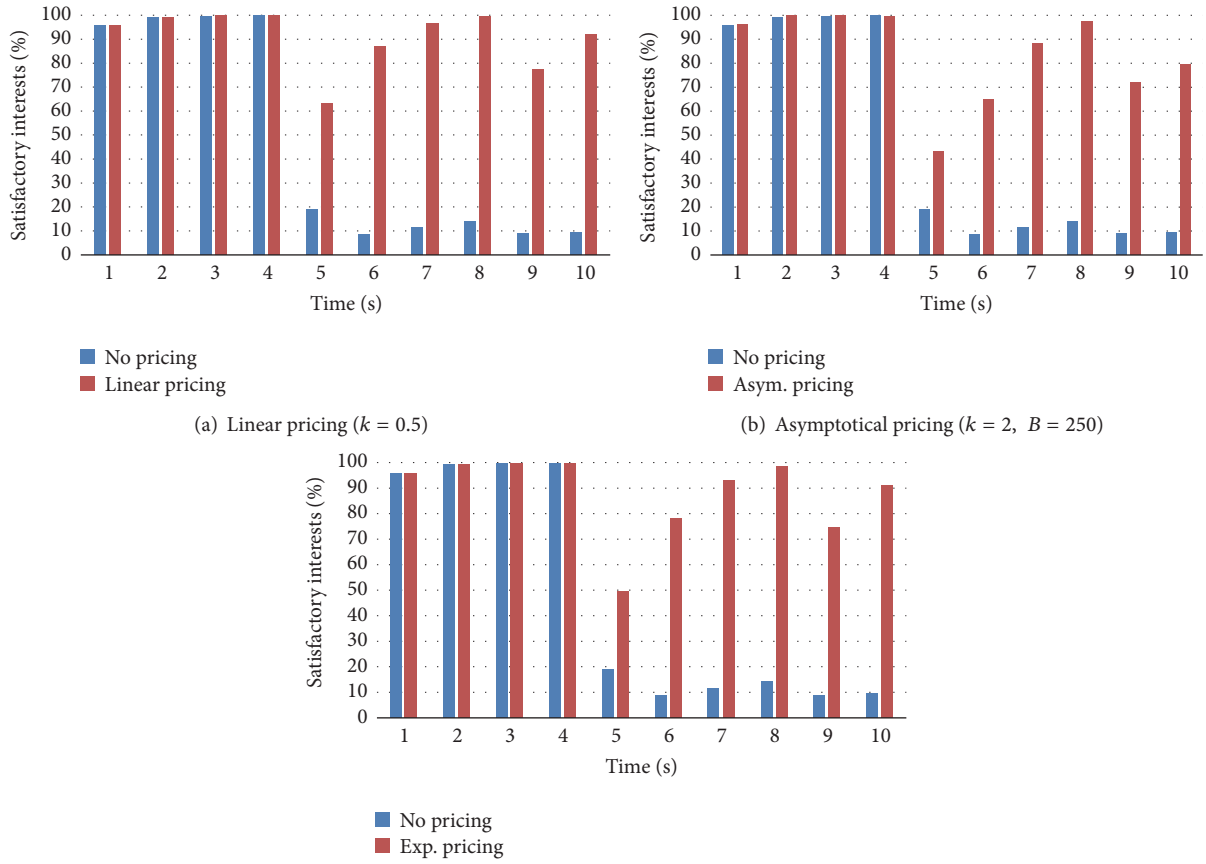
(c) Satisfactory interests

Figure 8: Simulation on exponential pricing strategy ($\alpha = 1$, $\beta = 0.02$).

Table 2: Utilization of pricing functions in NDN.

Pricing functions	Utilities/properties
Linear	Charging local interest request and PIT delay fee will increase linearly.
Asymptotic	PIT delay fee will become huge when PIT table reaches its limitations.
Exponential	The incremental PIT delay fee will increase exponentially.

down the concept of addressing and considering that many content packets will be cached in networking, the content producers cannot always fetch the real end users, and some NDN router node might be the last hop for forwarding interest request to content producers. Thus, the end users and the NDN routers have no sufficient prior knowledge to make proper prepayments to content producers. In fact, according to our abstraction of the proposed prototype, NDN router nodes need not consider the semantics of contents. Instead, NDN nodes just provide services of interests looking up and content delivering. In other words, NDN nodes play merely the role of logistics distribution, instead of the role of purchasing agents. Therefore, we suggest not to pay content producers. Moreover, in order to encourage NDN router nodes to perform better content delivering service, we can even ask content producers to pay NDN router nodes, and in return content producers can obtain what they deserved directly from the end users based on (post)accounting and auditing mechanisms. By doing so, another problem arises: How to protect content producers' benefits if a NDN router node sends many copies of some popular contents to many end users? Fortunately, this problem is essentially the issue of digital rights management (DRM) that has been studies extensively and there are a lot of mature solutions [26]. In other words, even if a NDN router node distributes many



(a) Linear pricing ($k = 0.5$)

(b) Asymptotical pricing ($k = 2$, $B = 250$)

(c) Exponential pricing ($\alpha = 1$, $\beta = 0.03$)

Figure 9: Simulation on a net-like topology.

copies of certain content, it merely gets multiple of content delivering fees, instead of the fee regarding the semantics and the quality of the content. If it charges more, it will face the risk of being detected and then have to afford punitive overcharging according to DRM or (post)auditing mechanisms.

4. Simulations and Evaluations

To verify the effectiveness of the proposed method, we conduct related simulations by using ndnSIM [16]. Our simulation is run over a PC workstation with 2.93 GHz CPU and 2 GB memory. The operation system is Windows 7, but the configurations and newly added specifications/functionalities of nsnSIM are implemented in Ubuntu that is running over a virtual machine created by VMware Workstation.

Our simulations are organized according to two different network topologies. The first is a very simple and tree-like topology that is merely used to illustrate our basic idea (see Figure 5(a)), while the second is a net-like topology that is randomly generated (see Figure 5(b)). For the first topology, there are in total 5 attack nodes (see grey nodes in Figure 5(a)) and they launch attack 5 seconds after the beginning of the corresponding simulations. For the second topology, we assume that all nodes behave normally at the beginning of the simulations, while after 4 seconds, 25 among them (i.e., about 15%) are randomly selected and specified as malicious. In both topologies, we, respectively, use linear pricing function, asymptotical pricing function, and exponential pricing function in charging PIT delay fee. In our simulations, the prepayment of an interest request is set to 100, and the maximum number of PIT items is set to 1000. Then, we collect related data and observe the evolution of not only the pricing function values, but also the numbers of unsatisfied interest requests in the related PIT tables (i.e., PIT item numbers) and the degree of satisfactory interest requests that is evaluated simply by the ratio of $n_s/(n_s + n_u)$, where n_s (resp., n_u) is the number of satisfied (resp., unsatisfied) interest requests.

Results are depicted in Figures 6, 7, 8, and 9, respectively.

(1) From Figures 6(a), 7(a), and 8(a), we can see different tendencies with different pricing functions. Note that in these pricing functions we always associate c in the related pricing functions with the number of interest requests coming from some ports, but based on our repeat testing we find that the results are a bit sensitive to other parameters like k, B, α, β, and so forth. In our simulations, we set these parameters based on the experience obtained from our earlier tests.

(2) From Figures 6(b), 7(b), and 8(b), we learn that on one hand, compared to the strategy without charging, these pricing functions are *indeed effective for keeping PIT tables from being quickly used out*; on the other hand, compared among these pricing functions, *the utility ratio of PIT tables with asymptotical pricing strategy is highest*, while the utility ratio of PIT tables with exponential pricing strategy is lowest.

(3) From Figures 6(c), 7(c), and 8(c), we learn that, compared to the strategy without charging, these pricing functions are *indeed effective for keeping high satisfactory ratio for newly coming interest requests on a long view*. But, this time, asymptotical pricing strategy does not manifest remarkable advantages over linear pricing strategy and exponential pricing strategy. In fact, the utility ratio of PIT tables and the satisfactory ratio for newly coming interest requests are interactions. To keep higher utility ratio of PIT tables means setting aside less room for newly coming interest requests and thus leading to lower satisfactory ratio. Therefore, we have to choose a balance between them. With this in mind, we think, as for the first simple topology, asymptotical pricing strategy outperforms the other two.

(4) However, from Figure 9, we can see that, as for the second topology, which is even close to real situations, asymptotical pricing strategy will lead to lowest satisfactory ratio for newly coming interest requests on a long view. Interestingly, linear pricing function outperforms the other two in this case. Again the proverb seems to be validated: *the simpler, the better.*

5. Summary and Future Work

An initial analysis of possibility of using economic levers in fighting interest flooding attacks (IFA) in Named Data Networking (NDN) is presented. We started by presenting a prototype for NDN that consists of seven basic business logics/steps, followed by an examination of compatibilities of existing micropayment systems and an analysis of utilization of some well-known pricing functions in NDN. Then, some basic simulations based on ndnSIM are developed and the results show that it is indeed effective for fighting IFA. Clearly, this is only the first step towards fighting DoS/DDoS in NDN with economic levers. More work is required to evaluate the effectiveness of the proposed prototype and to locate possible mismatched aspects of detailed business logics, such as the sensitiveness of different pricing functions with different setting on related parameters. Moreover, testbed-based, instead of simulation-based, experiments are needed for determining the real impacts of different micropayments and pricing functions on IFA in NDN.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Key Research and Development Program (no. 2016YFB0800602), the National Natural Science Foundation of China (NSFC) (nos. 61370194, 61502048), and the Engineering Planning Project of Communication University of China (no. 3132017XNG1720). The third author is also partially supported by JSPS KAKENHI Grant no. JP16K00117, KDDI Foundation.

References

[1] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "DoS and DDoS in named data networking," in *Proceedings of the 2013 IEEE 2013 22nd International Conference on Computer Communication and Networks, ICCCN 2013*, bhs, August 2013.

[2] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system," *ACM SIGOPS Operating Systems Review*, vol. 33, no. 5, pp. 186–201, 1999.

[3] N. Tolia, M. Kaminsky, and D. Andersen, "An architecture for Internet data transfer," in *Proceedings of the 3rd conference on Networked Systems Design Implementation (NSDI*, pp. 253–266, 2006.

[4] T. Koponen, M. Chawla, B.-G. Chun et al., "A data-oriented (and beyond) network architecture," in *Proceedings of the ACM SIGCOMM 2007: Conference on Computer Communications*, pp. 181–192, jpn, August 2007.

[5] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker, "ROFL: routing on flat labels," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 363–374, 2006.

[6] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 205–218, 2004.

[7] Project PSIRP. http://www.psirp.org, 2010.

[8] V. Jacobson, "Special plenary invited short course: (CCN) Content-centric networking," in *Future Internet Summer School*, Germany, Bremen, 2009.

[9] Project CCNx. http://www.ccnx.org, 2011.

[10] V. Jacobson, D. Smetters K, J. Thorton D et al., "Networking named content," *Communications of the ACM*, vol. 55, no. 1, pp. 117–124, 2012.

[11] A. Juels and J. Brainard, "Client puzzles: a cryptographic defense against connection depletion attacks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 151–165, 1999.

[12] V. Jacobson, D. K. Smetters, N. H. Briggs et al., "VoCCN: Voice-over content-centric networks," in *Proceedings of the 2009 workshop on Re-architecting the internet*, pp. 1–6, 2009.

[13] L. Zhang and D. Estrin, "Named data networking (NDN," Tech. Rep., 2010.

[14] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, "ACT: audio conference tool over named data networking," in *Proceedings of the 2011 ACM SIGCOMM Workshop on Information-Centric Networking, ICN 2011, Co-located with SIGCOMM 2011*, pp. 68–73, 2011.

[15] H. Yuan and P. Crowley, "Experimental evaluation of content distribution with NDN and HTTP," in *Proceedings of the IEEE INFOCOM 2013 Mini-Conference*, pp. 240–244, 2013.

[16] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM, NDN simulator for NS-3," Tech. Rep., 2012.

[17] A. Atanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *Proceedings of the IFIP Networking*, pp. 1–9, 2013.

[18] D. Mankins, R. Krishnan, C. Boyd, J. Zao, and M. Frentz, "Mitigating distributed denial of service attacks with dynamic resource pricing," in *Proceedings of the 17th Annual Computer Security Applications Conference, ACSAC 2001*, pp. 411–421, usa, December 2001.

[19] M. Abadi, M. Burrows, M. Manasse, and T. Wobber, "Moderately hard, memory-bound functions," in *Proceedings of the 10th Annual Network and Distributed System Security Symposium (NDSS)*, pp. 25–39, Internet Society, 2003.

[20] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO)*, pp. 139–147, Springer-Verlag, London, UK, 1992.

[21] Y. Shen, F. Fan, W. Xie, and L. Mo, "Re-Traffic pricing for fighting against DDoS," in *Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM)*, pp. 332–336, IEEE Computer Society, Washington, DC, USA, 2008.

[22] R. Rivest and A. Shamir, "PayWord and MicroMint: Two Simple Micro-payment Schemes, Proceeding of the Security Protocols Workshop," *Lecture Notes in Computer Science*, vol. 1189, pp. 69–87, 1997.

[23] S. C. Glassman, M. S. Manasse, M. Abadi, P. Gauthier, and P. Sobalvarro, "The Millicent protocol for inexpensive electronic commerce," *World Wide Web*, vol. 1, no. 1, 1996, https://www.w3.org/Conferences/WWW4/Papers/246/.

[24] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org.

[25] S. Khanna, S. S. Venkatesh, O. Fatemieh, F. Khan, and C. A. Gunter, "Adaptive selective verification: An efficient adaptive countermeasure to thwart DoS attacks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 715–728, 2012.

[26] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, "Digital rights management for content distribution," in *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 (ACSW Frontiers 2003)*, vol. 21, pp. 49–58, Australian Computer Society, 2003.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4070616, 12 pages
https://doi.org/10.1155/2017/4070616

Research Article

Research on Ciphertext-Policy Attribute-Based Encryption with Attribute Level User Revocation in Cloud Storage

Guangbo Wang and Jianhua Wang

Zhengzhou Information Science and Technology Institute, Zhengzhou, Henan 450004, China

Correspondence should be addressed to Guangbo Wang; 691759571@qq.com

Received 17 February 2017; Revised 1 April 2017; Accepted 5 April 2017; Published 23 May 2017

Academic Editor: Liu Yuhong

Attribute-based encryption (ABE) scheme is more and more widely used in the cloud storage, which can achieve fine-grained access control. However, it is an important challenge to solve dynamic user and attribute revocation in the original scheme. In order to solve this problem, this paper proposes a ciphertext-policy ABE (CP-ABE) scheme which can achieve attribute level user attribution. In this scheme, if some attribute is revoked, then the ciphertext corresponding to this attribute will be updated so that only the individuals whose attributes meet the access control policy and have not been revoked will be able to carry out the key updating and decrypt the ciphertext successfully. This scheme is proved selective-structure secure based on the q-Parallel Bilinear Diffie-Hellman Exponent (BDHE) assumption in the standard model. Finally, the performance analysis and experimental verification have been carried out in this paper, and the experimental results show that, compared with the existing revocation schemes, although our scheme increases the computational load of storage service provider (CSP) in order to achieve the attribute revocation, it does not need the participation of attribute authority (AA), which reduces the computational load of AA. Moreover, the user does not need any additional parameters to achieve the attribute revocation except for the private key, thus saving the storage space greatly.

1. Introduction

With the advent of big data era, there is an increasing number of user data. In order to achieve the sharing of data and reduce the cost at the same time, using the third party, namely, cloud storage provider (CSP), will be an excellent priority. The cloud storage, which emerged as the extension and development of cloud computing, achieves the function that the users can access the data conveniently at any time and at any place by any networking equipment; therefore, it has been more and more extensively used. However, the users' data are stored in the CSP and got rid of the users' actual control; therefore, how to guarantee the users privacy and data security as much as possible without reducing the quality of service has become a key problem of secure cloud storage.

Sahai and Waters in 2005 proposed the notation of attribute-based encryption (ABE) [1] in which the ciphertext and key are, respectively, associated with a series of attributes, and an access structure is specified to define the attribute set that can be used to decrypt the ciphertext successfully. ABE can achieve fine-grained access control by using the flexible access structure, so it has been widely used in the cloud storage. The initial ABE schemes can only achieve the threshold operations so that the policy expression is not rich enough. To solve this problem, some scholars have proposed the ciphertext-policy ABE (CP-ABE) mechanism [2–4] and key-policy ABE (KP-ABE) mechanism [5, 6], which can realize rich attribute operations so as to support flexible access control policy.

However, the application of ABE in cloud storage also brings serious security challenges. There are a large number of users in the cloud storage environment, and different users may share the same attribute in the application of ABE. Therefore, if some attribute of a user is revoked, how to recall the user's corresponding access permissions without affecting the normal access of other legitimate users and posing a large load on the system has become an urgent problem to

be solved. Therefore, this paper mainly pursues the relative research on this issue.

Recently, individuals pay more and more attention to the problem of user revocation in the practical application of ABE. Ostrovsky et al. proposed an ABE scheme with system level user revocation [7]. In this scheme, the revocation is carried out by implementing the "NOT" operation on "AND" gates; however, the efficiency is rather low.

Subsequently, Staddon et al. proposed a KP-ABE scheme [8] which can achieve the revocation of users; however, this scheme is limited to be used if and only if the number of attributes associated with ciphertext is just half of the whole attributes in the system; therefore, the limit is too high which impedes its actual application. Liang et al. proposed a CP-ABE scheme [9] which achieved the revocation by using a binary tree. In this scheme, an attribute authority is responsible for generating the updating key for implementing the revocation; however, the efficiency is also very low. Moreover, it increases the computation and communication burden on the attribute authority greatly which may become the bottleneck. In addition, all the above schemes can only achieve the system level user revocation; namely, once some attribute of a user is revoked, he will lose not only the access permission corresponding to the revoked attribute but also the access permissions corresponding to the other legitimate attributes.

In the aspect of attribute revocation, individuals in the literatures [10–12] strove to achieve the revocation by setting the validity period for each attribute. This method is called coarse-grained revocation because it cannot realize the timely revocation. To solve this problem, Hur and Noh proposed a novel CP-ABE scheme in the literature [13] to realize the revocation by using a key encryption key tree, which can also achieve attribute level user revocation; namely, the revocation to some attribute of a user cannot affect the normal access of other legitimate attributes. In this scheme, if an attribute is revoked, then the CSP will generate a new key encryption key and reencrypts the ciphertext. However, each user needs to store $\log(n_u + 1)$ key encryption keys additionally, where n_u denotes the number of all the users in this scheme. Moreover, the scheme is proved to be secure in the generic group model which possesses heuristic security rather than provable security; therefore, some schemes proved secure in the generic group model are found to be unsafe in practical application. Subsequently, Yang et al. proposed a CP-ABE scheme [14] in the environment of cloud storage. In this scheme, the attribute authority generates two corresponding public parameters for each attribute, and once the revocation is implemented, the attribute authority needs to update the public parameters for the revoked attribute and the secret key for the user, which increases not only the computation load on the attribute authority but also the communication load between the attribute authority and the user.

In this paper, we propose a CP-ABE scheme that combines proxy reencryption methods to achieve the revocation. In this scheme, we achieve the revocation with the help of CSP, which offloads most of revocation operations for the attribute authority that has limited resources. If some attribute is revoked, then the ciphertext corresponding to this attribute will be updated by the CSP so that only the users whose attributes meet the access control policy and have not been revoked will be able to carry out the key updating and decrypt the ciphertext successfully. Additionally, in this scheme, we achieve the fine-grained attribute level user revocation; namely, the revocation to an attribute of some user cannot affect the normal access of this user's other legitimate attributes. Finally, we carry out the performance analysis and experimental verification to demonstrate the characteristics, which shows that, compared with the existing revocation schemes, although our scheme increases the computational load of CSP in order to achieve the attribute revocation, it does not need the participation of AA. Moreover, the user does not need any additional parameters to achieve the attribute revocation except of the private key, thus saving the storage space greatly.

2. Preliminaries

Before proposing the concrete scheme in this paper, we first introduce the related technologies that will be used including bilinear group, linear secret-sharing scheme (LSSS), and deterministic q-Parallel Bilinear Diffie-Hellman Exponent (BDHE) assumption.

2.1. Bilinear Map. In this part, we will briefly take a view to several facts related to the bilinear group as follows.

Definition 1 (bilinear map). The bilinear group has been widely used in various cryptographic systems after it was proposed for the first time. Let ψ be a group parameters generation algorithm which takes as input the security parameter λ and outputs the group parameters $(p, \mathbb{G}, \mathbb{G}_T, e)$. In these group parameters, p denotes a big prime whose size is determined by the security parameter λ, $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups with order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map satisfying the following properties:

(1) Bilinearity: $\forall u, v \in \mathbb{G}$, $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.

(2) Nondegeneracy: $\exists g \in \mathbb{G}$ satisfying that $e(g, g)$ has order p in $\mathbb{G}_T$.

(3) Computability: there exists an efficient algorithm to compute the bilinear pairing.

2.2. Linear Secret-Sharing Scheme

Definition 2 (linear secret-sharing scheme (LSSS) [15]). A secret-sharing scheme $\prod$ over a set of parties $\mathcal{P}$ is a LSSS (over $\mathbb{Z}_p$) if it satisfies the following properties:

(1) The secret share of each party constitutes a vector over $\mathbb{Z}_p$.

(2) For each secret-sharing scheme $\prod$, there exists a share-generation matrix $\mathbf{M}(l \times n)$ where, for each row $\mathbf{M}_i$ of the matrix $\mathbf{M}$, we define a function ρ: $\{1, \ldots, l\} \to \mathcal{P}$ that maps it to the corresponding party $\rho(i)$. Considering a vector $\vec{v} = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the sharing secret and parameters $r_2, \ldots, r_n \in \mathbb{Z}_p$ are chosen randomly to conceal the secret, then $\mathbf{M}\vec{v}$ is a vector that is composed of l shares of

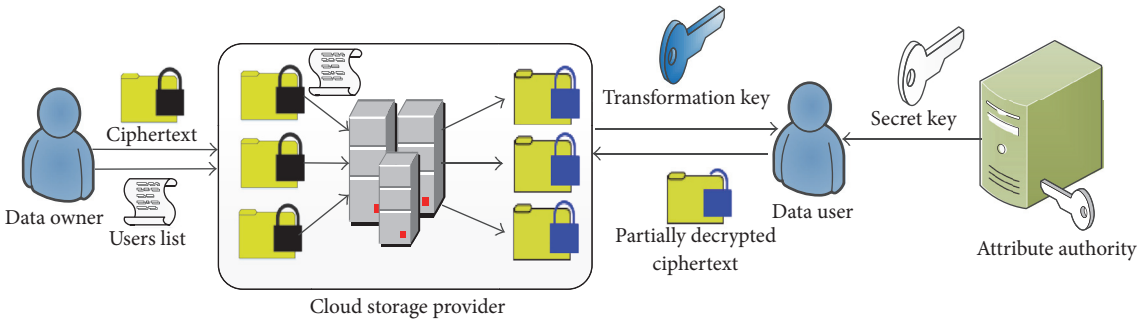


Figure 1: System model.

the secret s. Moreover, $\lambda_i = (\mathbf{M}\vec{v})_i$ denotes the secret share possessed by the party $\rho(i)$.

Suppose $\prod$ is a LSSS for the access structure $(\mathbf{M}, \rho)$ and S denotes any authorized set for $(\mathbf{M}, \rho)$. We define the set $\mathscr{D} \subset \{1, 2, \ldots, l\}$ as $\mathscr{D} = \{i: \rho(i) \in S\}$; then, the constants $\{w_i \in \mathbb{Z}_p\}_{i\in\mathscr{D}}$ can be computed in polynomial time such that if $\{\lambda_i\}$ are valid shares of any secret s according to $\prod$, then we have $\sum_{i\in\mathscr{D}} w_i\lambda_i = s$.

2.3. Decisional q-Parallel Bilinear Diffie-Hellman Exponent Assumption

Definition 3 (q-parallel BDHE assumption [16]). Let $\mathbb{G}$ denote the bilinear group with prime order p, the parameters $a, s, b_1, \ldots, b_q$ are chosen randomly in $\mathbb{Z}_p$, and g is a generator of $\mathbb{G}$. Then, the decisional q-Parallel BDHE assumption is that if there is an attacker $\mathscr{A}$ who is given the parameters

$$\begin{aligned}
\vec{y} &= g, g^s, g^a, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}},\\
\vec{y} &= g^{sb_j}, g^{a/b_j}, \ldots, g^{a^q/b_j}, \ldots, g^{a^{2q}/b_j}, \quad \forall_{1\le j\le q},\\
\vec{y} &= g^{asb_k/b_j}, \ldots, g^{a^q sb_k/b_j} \quad \forall_{1\le k,j\le q},
\end{aligned} \tag{1}$$

then, it is hard for $\mathscr{A}$ to distinguish $e(g, g)^{a^{q+1}s}$ from a random element in $\mathbb{G}_T$. In addition, a polynomial time algorithm $\mathscr{B}$ will use the output of $\mathscr{A}$ to make a guess, and we define the advantage of $\mathscr{B}$ to solve the q-Parallel BDHE assumption in $\mathbb{G}$ and $\mathbb{G}_T$ as

$$\left|\Pr\left[\mathscr{B}\left(\vec{y}, e\left(g, g\right)^{a^{q+1}s}\right) = 0\right] - \Pr\left[\mathscr{B}\left(\vec{y}, R\right) = 0\right]\right|. \tag{2}$$

If there is no polynomial time algorithm to solve the q-Parallel BDHE assumption with a nonnegligible advantage, then we can say that the assumption holds in $\mathbb{G}$ and $\mathbb{G}_T$.

3. Attribute-Based Encryption

In this part, we will first give the system model for our proposed CP-ABE scheme with attribute level user revocation, and then we give a selectively secure model in terms of the ciphertext indistinguishability under a chosen plaintext attack (IND-CPA) [17] which is defined between a polynomial time attacker $\mathscr{A}$ and challenger $\mathscr{B}$. Finally, we will give the detailed construction.

3.1. System Model.

The concrete system model of our proposed CP-ABE scheme is shown as in Figure 1, which mainly consists of four entities as follows.

(1) Attribute Authority (AA). It is responsible for implementing the system setup algorithm to generate the system parameters and implementing the key generating algorithm to generate the secret key for the data user.

(2) Data Owner (DO). He is responsible for implementing the data encryption algorithm on the plaintext data and sends the generated ciphertext to the CSP. If the DO decides that some attribute needs to be revoked, he will first designate the responding revoked users list and then send the list to the CSP.

(3) Data User (DU). He is responsible for implementing the decryption algorithm. If the DU wants to access the data in the CSP, he will first send his transformation key to the CSP for partial decryption. Once the DU receives the partially decrypted ciphertext, he will use his secret key to implement the final decryption.

(4) Cloud Storage Provider (CSP). He is responsible for implementing the data reencryption algorithm to achieve the ciphertext updating and implementing the partial decryption algorithm for the DU. Here, we assume that the CSP is curious but honest; namely, he will honestly execute the tasks assigned by other legitimate entities in the system; however, he has the incentive to learn the contents of encrypted data as much as possible.

3.2. Selectively Secure Model.

This security model mainly draws lessons from the technique proposed by Tu et al. in the literature [18]. In this model, the attacker $\mathscr{A}$ firstly needs to submit a challenge access structure and a revocation list, and as a response he will obtain the corresponding public key parameters. Subsequently, $\mathscr{A}$ begins to make a series of secret key queries and ciphertext reencryption queries. In the challenge phase, $\mathscr{A}$ will give two messages with the equal length, and then the challenger $\mathscr{B}$ chooses to encrypt one of these two messages based on the random sampling. Next, $\mathscr{A}$ continues to make the secret key query and ciphertext reencryption query and finally outputs a random guess. If the guess is correct, then we can say $\mathscr{A}$ wins the game. The specific definition of this security model is given as follows.

Init. The attacker $\mathcal{A}$ initially chooses the challenge access control structure $\mathbb{A}^*$ and the revocation users list RL_{x^*} of attribute x^*.

Setup. The challenger $\mathcal{B}$ runs the algorithm *Setup* to obtain the public key PK and the master key MK. Finally, $\mathcal{B}$ gives PK to the attacker $\mathcal{A}$ and keeps MK private to itself.

Query Phase 1. The attacker $\mathcal{A}$ adaptively makes a series of secret key queries corresponding to the identity-attribute tuple, namely, $(\mathrm{ID}_1, S_1), \ldots, (\mathrm{ID}_{q_1}, S_{q_1})$; if $\mathrm{ID}_i \notin \mathrm{RL}_{x^*}$, then we set $S_i' = S_i$; otherwise, we set $S_i' = S_i/\{x^*\}$. Note that it must satisfy the restriction that any attributes set S_i' cannot satisfy the challenge access control structure $\mathbb{A}^*$ in this phase. In addition, $\mathcal{A}$ can also make a series of ciphertext reencryption queries associated with the revocation users list of some attribute and the ciphertext.

Challenge. The attacker $\mathcal{A}$ outputs two messages m_0 and m_1 with the equal length to the challenger $\mathcal{B}$. Then, $\mathcal{B}$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts the message m_β under the access control structure $\mathbb{A}^*$ to generate the ciphertext CT^*. Finally, $\mathcal{B}$ sends CT^* to $\mathcal{A}$ as the challenge ciphertext.

Query Phase 2. The attacker $\mathcal{A}$ continues to make a series of secret key queries and ciphertext reencryption queries as in *Query Phase 1* with the same restriction.

Guess. The attacker $\mathcal{A}$ outputs its guess β' for β, and if $\beta' = \beta$, then $\mathcal{A}$ wins the game. In addition, the advantage of $\mathcal{A}$ in this game is defined as $\mathrm{Adv}_{\mathcal{A}} = |\Pr[\beta' = \beta] - 1/2|$.

If there is no polynomial time algorithm to break the security model above with a nonnegligible advantage, then we can say that our proposed CP-ABE scheme with attribute level user revocation is secure.

3.3. Construction. In this part, we will give the concrete construction of our proposed CP-ABE scheme. In our scheme, the attribute authority will first generate the system parameters that will be used in the subsequent algorithms. If the data owner DO wants to store his data on the CSP, he will first encrypt the data with some access control policy to generate the corresponding ciphertext, then he will send the ciphertext to the CSP. Once the DO decides that an attribute of some users list needs to be revoked, he will send the users list to the CSP. Then, the CSP will implement the reencryption on the ciphertext so that only the users whose attributes meet the access control policy associated with the ciphertext and have not been revoked will be able to carry out the key updating and decrypt the ciphertext successfully. In addition, we use the outsourcing decryption to improve the efficiency; namely, the data user (DU) can send his transformation key to the CSP for partial decryption, which makes full use of the computing resources in the CSP. Once the DU gets the partially decrypted ciphertext, he will implement the final decryption faster with less computing resources.

3.3.1. System Setup. In this phase, the attribute authority will generate the corresponding system parameters including the public key and the master key. The public key is accessible by all the entities in the system and the master key is kept private to the attribute authority.

(1) Setup ($\mathrm{setup}(\lambda, U, n) \rightarrow (\mathrm{PK}, \mathrm{MK})$). The setup algorithm takes as input the security parameter λ, the attributes set U, and the number n of users in the system; then, it runs the group parameters generation function ψ to obtain $(\mathbb{G}, \mathbb{G}_T, p, e)$, where p denotes a big prime, $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups with order p, and e is a bilinear map. Let g be the generator of $\mathbb{G}$. Then, the algorithm chooses random exponents $\alpha, \beta \in \mathbb{Z}_p$ and sets $g_i = g^{(\alpha^i)} \in \mathbb{G}$, where $i = 1, 2, \ldots, n, n+2, \ldots, 2n$. Next, it chooses a random exponent $\gamma \in \mathbb{Z}_p$ and sets $v = g^\gamma$. For each attribute $i \in U$, the algorithm chooses random parameters $h_i \in \mathbb{G}$. Finally, the system public key PK is set as $\mathrm{PK} = (p, g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v, e(g, g)^\beta, h_1, \ldots, h_U)$ and the master key MK is set as $\mathrm{MK} = (\alpha, \gamma, g^\beta)$.

3.3.2. Data Encryption. If the data owner wants to store his data $m \in \mathbb{G}_T$ on the CSP, then he will first define an access control policy $(\mathbf{M}, \rho)$ where $\mathbf{M}$ is a $l \times n$ matrix, and the function ρ maps each row $\mathbf{M}_i$ of $\mathbf{M}$ to one corresponding attribute $\rho(i)$ with the restriction that ρ cannot map two distinct rows to one attribute just as in literature [19]. Next, the data encryption algorithm runs $\mathrm{Encrypt}(\mathrm{PK}, m, (\mathbf{M}, \rho))$ to encrypt the data m. Note that the encryption on the data m needs to multiply it with some group element in $\mathbb{G}_T$; therefore, m is also defined as an element in $\mathbb{G}_T$. If we want to encrypt some arbitrary data, then we can define a hash function: $H : \mathbb{Z}_p \rightarrow \mathbb{G}_T$ which maps the arbitrary data to an element in the group $\mathbb{G}_T$.

(2) Encrypt ($\mathrm{encrypt}(\mathrm{PK}, m, (\mathbf{M}, \rho)) \rightarrow \mathrm{CT}$). The encryption algorithm takes as input the public key PK, the plaintext message m, and an access control policy $(\mathbf{M}, \rho)$; then, it chooses random parameters $s, v_2, \ldots, v_n \in \mathbb{Z}_p$ and defines the vector $\mathbf{v} = (s, v_2, \ldots, v_n)$. For each row $\mathbf{M}_i$ of $\mathbf{M}$, the algorithm computes the inner product $\lambda_i = \mathbf{M}_i \cdot \mathbf{v}$, and then it chooses a random exponent $r_i \in \mathbb{Z}_p$ and outputs the ciphertext as follows:

$$\begin{aligned} \mathrm{CT} = \Big(& (\mathbf{M}, \rho),\ C = m \cdot e(g, g)^{\beta s},\ C_0 \\ & = g^s,\ \left\{ C_{i,1} = g_1^{\lambda_i} h_{\rho(i)}^{-r_i},\ C_{i,2} = g^{r_i} \right\}_{i=1}^{l} \Big). \end{aligned} \tag{3}$$

3.3.3. Data Reencryption. If the DO decides that the attribute x of users list RL_x needs to be revoked, then he will send (x, RL_x) to the CSP. Once the CSP receives (x, RL_x), he will use the broadcast encryption to update the ciphertext for the purpose of revoking the access permission corresponding to attribute x without affecting the normal access of other legitimate attributes for the users in RL_x.

(3) Re-Encrypt ($\text{Re-encrypt}(\mathrm{PK}, \mathrm{CT}, \mathrm{RL}_x) \rightarrow \mathrm{CT}''$). The reencryption algorithm takes as input the public key PK, the ciphertext $\mathrm{CT} = (C, C_0, \{C_{i,1}, C_{i,2}\}_{i=1}^{l})$, and the revocation

users list RL_x, and then it chooses a random exponent $v_x \in \mathbb{Z}_p^*$ and outputs the reencrypted ciphertext as follows:

$$\begin{aligned}\mathrm{CT}' = \Big(& (\mathbf{A}, \rho),\ C' = C,\ C_0' = C_0,\ \rho(i) \neq x\colon C_{i,1}' \\ & = C_{i,1},\ C_{i,2}' = C_{i,2},\ \rho(i) = x\colon C_{x,1}' = C_{x,1},\ C_{x,2}' \\ & = \left(C_{x,2}\right)^{1/v_x} = \left(g^{r_x}\right)^{1/v_x} \Big).\end{aligned} \tag{4}$$

Next, the algorithm chooses random parameters $\widetilde{s}, \widetilde{v}_2, \ldots, \widetilde{v}_n \in \mathbb{Z}_p$ and defines the vector $\widetilde{\mathbf{v}} = (\widetilde{s}, \widetilde{v}_2, \ldots, \widetilde{v}_n)$. Note that the reencryption algorithm will use the same access control policy $(\mathbf{M}, \rho)$ as in the *Encrypt* algorithm. For each row $\mathbf{M}_i$ of the matrix $\mathbf{M}$, it computes the inner product $\widetilde{\lambda}_i = \mathbf{M}_i \cdot \widetilde{\mathbf{v}}$ and chooses a random exponent $\widetilde{r}_i \in \mathbb{Z}_p$. Then, the algorithm defines a broadcast users set $N = n \setminus \{\mathrm{RL}_x\}$ and outputs the ciphertext header generated by encrypting the exponent v_x as follows:

$$\begin{aligned}\mathrm{Hdr}_x = \Bigg(& \mathrm{RL}_x,\ \widetilde{C} = v_x \cdot e\left(g_n, g_1\right)^{\widetilde{s}},\ \widetilde{C}_0 = g^{\widetilde{s}},\ \widetilde{C}_1 \\ & = \left(v \left(\prod_{j \in N} g_{n+1-j} \right)^{-1} \right)^{\widetilde{s}}, \\ & \left\{ \widetilde{C}_{i,1} = \left(g_1\right)^{\widetilde{\lambda}_i} h_{\rho(i)}^{-\widetilde{r}_i},\ \widetilde{C}_{i,2} = g^{\widetilde{r}_i} \right\}_{i=1}^{l} \Bigg).\end{aligned} \tag{5}$$

Finally, it returns the ciphertext as $\mathrm{CT}'' = (\mathrm{CT}', \mathrm{Hdr}_x)$.

3.3.4. Key Generation. In order to improve the decryption efficiency, we outsource the decryption of ciphertext to the CSP that has plenty of computing resources. The concrete key generation algorithm is given as follows.

(4) KeyGen ($\mathrm{keygen}_{\mathrm{out}}(\mathrm{PK}, \mathrm{MK}, \mathrm{ID}, \mathrm{S}) \rightarrow \mathrm{SK}$). The key generation algorithm takes as input the public key PK, the master key MK, a user's identity ID, and the attributes set S, and then it chooses a random exponent $r' \in \mathbb{Z}_p$ and generates the corresponding key $\mathrm{SK}' = (K', \widetilde{K}', L', \{K_i'\}_{i \in \mathrm{S}})$, where

$$\begin{aligned} K' &= g^{\alpha^{\mathrm{ID}} \gamma} g^{\alpha r'}, \\ \widetilde{K}' &= g^{\beta} g^{\alpha r'}, \\ L' &= g^{r'}, \\ \left\{ K_i' = h_i^{r'} \right\}_{i \in S}. & \end{aligned} \tag{6}$$

Next, the algorithm continues to choose a random exponent $z \in \mathbb{Z}_p^*$ and computes

$$\begin{aligned} K &= \left(K'\right)^{1/z} = \left(g^{\alpha^{\mathrm{ID}} \gamma}\right)^{1/z} \left(g^{\alpha r'}\right)^{1/z}, \\ \widetilde{K} &= \widetilde{K}'^{1/z} = \left(g^{\beta}\right)^{1/z} \left(g^{\alpha r'}\right)^{1/z}, \\ L &= \left(L'\right)^{1/z} = \left(g^{r'}\right)^{1/z}, \\ \left\{ K_i = \left(K_i'\right)^{1/z} = \left(h_i^{r'}\right)^{1/z} \right\}_{i \in S}. & \end{aligned} \tag{7}$$

Let $r = r'/z$; then, we have

$$\begin{aligned} K &= \left(g^{\alpha^{\mathrm{ID}} \gamma}\right)^{1/z} g^{\alpha r}, \\ \widetilde{K} &= \left(g^{\beta}\right)^{1/z} g^{\alpha r}, \\ L &= g^{r}, \\ \{K_i = h_i^{r}\}_{i \in S}. & \end{aligned} \tag{8}$$

Finally, we set the outsourced transformation key as $\mathrm{TK} = (K, \widetilde{K}, L, \{K_i = h_i^r\}_{i \in S})$ and the secret key as $\mathrm{SK} = (z, \mathrm{TK})$.

3.3.5. Partial Decryption. In order to achieve the outsourced decryption, the user needs to send his transformation key TK to the CSP. Note that the transformation key cannot leak any useful information associated with the secret key SK and the plaintext data m. The concrete partial decryption algorithm is given as follows.

(5) Transform ($\mathrm{transform}_{\mathrm{out}}(\mathrm{TK}, \mathrm{CT}'') \rightarrow \mathrm{TCT}$). The transformation algorithm takes as input the transformation key $\mathrm{TK} = (K, \widetilde{K}, L, \{K_i = h_i^r\}_{i \in \mathrm{S}})$ and the ciphertext $\mathrm{CT}'' = (\mathrm{CT}', \mathrm{Hdr}_x)$.

(1) If there is no attribute revoked, namely, $\mathrm{Hdr}_x = \Phi$, then we have the following.

Here, we have $\mathrm{CT}'' = ((\mathbf{M}, \rho), C, C_0, \{C_{i,1}, C_{i,2}\}_{i=1}^{l})$, and if the attributes set S associated with TK satisfies the access control policy $(\mathbf{M}, \rho)$ included in CT'', then the CSP computes the values $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ satisfying $\sum_{i \in I} w_i \mathbf{M}_i = (1, 0, \ldots, 0)$ in polynomial time. Next, it computes

$$\begin{aligned} B &= \prod_{i \in I} e\left(C_{i,1}, L\right)^{w_i} e\left(C_{i,2}, K_{\rho(i)}\right)^{w_i} \\ &= \prod_{i \in I} e\left(g_1^{\lambda_i} h_{\rho(i)}^{-r_i}, g^{r}\right)^{w_i} e\left(g^{r_i}, h_{\rho(i)}^{r}\right)^{w_i} = e(g, g)^{\alpha r s}, \\ D &= e\left(C_0, \widetilde{K}\right) = e\left(g^{s}, g^{\beta/z} g^{\alpha r}\right) \\ &= e(g, g)^{\beta s / z} e(g, g)^{\alpha r s}, \\ E &= \frac{D}{B} = \frac{e(g, g)^{\beta s / z} e(g, g)^{\alpha r s}}{e(g, g)^{\alpha r s}} = e(g, g)^{\beta s / z}. \end{aligned} \tag{9}$$

Once the partial decryption is over, the CSP sends TCT = (C, E) to the corresponding user for the final decryption.

(2) If the attribute x of users list RL_x is revoked, namely, $\mathrm{Hdr}_x \neq \Phi$, then we have the following.

Here, we have $\mathrm{CT}' = ((\mathbf{M}, \rho), C', C_0', \{C_{i,1}', C_{i,2}'\}_{i=1}^{l})$ and $\mathrm{Hdr}_x = (\mathrm{RL}_x, \widetilde{C}, \widetilde{C}_0, \widetilde{C}_1, \{\widetilde{C}_{i,1}, \widetilde{C}_{i,2}\}_{i=1}^{l})$, and if the attributes set S satisfies the access control policy $(\mathbf{M}, \rho)$ and $\mathrm{ID} \notin \mathrm{RL}_x$, then the CSP implements the partial decryption on the ciphertext header Hdr_x. It also computes the values $\{\widetilde{w}_i \in \mathbb{Z}_p\}_{i\in\tilde{I}}$ satisfying $\sum_{i\in\tilde{I}} \widetilde{w}_i \mathbf{M}_i = (1, 0, \ldots, 0)$ and then continues to compute

$$\begin{aligned}
B_x &= \prod_{i\in\tilde{I}} e\left(\widetilde{C}_{i,1}, L\right)^{\widetilde{w}_i} e\left(\widetilde{C}_{i,2}, K_{\rho(i)}\right)^{\widetilde{w}_i} \\
&= \prod_{i\in\tilde{I}} e\left(g_1^{\tilde{\lambda}_i} h_{\rho(i)}^{-\tilde{r}_i}, g^r\right)^{\widetilde{w}_i} e\left(g^{-\tilde{r}_i}, h_{\rho(i)}^r\right)^{\widetilde{w}_i} \\
&= e(g,g)^{\alpha r s'}, \\
D_x &= e\left(\widetilde{C}_0, K\right) = e(g,g)^{\alpha^{\mathrm{ID}} \gamma \tilde{s}/z} e(g,g)^{\alpha r \tilde{s}}, \\
E_x &= \frac{D_x}{B_x} = e(g,g)^{\alpha^{\mathrm{ID}} \gamma \tilde{s}/z}, \\
F_x &= \frac{e\left(g_{\mathrm{ID}}, \widetilde{C}_1\right)}{e\left(\prod_{\substack{j\in N \\ j\neq \mathrm{ID}}} g_{n+1-j+\mathrm{ID}}, \widetilde{C}_0\right)^{-1}} \\
&= \frac{e\left(g_{\mathrm{ID}}, \left(v\left(\prod_{j\in N} g_{n+1-j}\right)^{-1}\right)^{\tilde{s}}\right)}{e\left(\prod_{\substack{j\in N \\ j\neq \mathrm{ID}}} g_{n+1-j+\mathrm{ID}}, g^{\tilde{s}}\right)^{-1}} \\
&= e\left(g_{\mathrm{ID}}, v\right)^{\tilde{s}} \cdot e\left(g_{n+1}, g\right)^{-\tilde{s}}.
\end{aligned} \tag{10}$$

Therefore, the partially decrypted ciphertext header is set as $\mathrm{Hdr}_x' = (\widetilde{C}, E_x, F_x)$.

Next, the CSP implements the partial decryption on the ciphertext CT' as follows:

$$\begin{aligned}
B_i &= e\left(C_{i,1}', L\right) e\left(C_{i,2}', K_{\rho(i)}\right) = e(g,g)^{\alpha r \lambda_i} \quad \rho(i) \neq x, \\
D &= e\left(C_0', \widetilde{K}\right) = e\left(g^s, g^{\beta/z} g^{\alpha r}\right) \\
&= e(g,g)^{\beta s/z} e(g,g)^{\alpha r s} \quad \rho(i) = x.
\end{aligned} \tag{11}$$

Therefore, the partially decrypted ciphertext is set as

$$\begin{aligned}
\mathrm{TCT}' = \Big((\mathbf{M}, \rho),\ C' = m \\
\cdot e(g,g)^{\beta s}, \{B_i\}_{\rho(i)\neq x}, C_{x,1}', C_{x,2}', D\Big).
\end{aligned} \tag{12}$$

Once the partial decryption is over, the CSP sends $\mathrm{TCT} = (\mathrm{TCT}', \mathrm{Hdr}_x')$ to the corresponding user for the final decryption.

3.3.6. Decryption. Once the user gets the partially decrypted ciphertext, he will use his secret key to implement the final decryption for obtaining the plaintext message as follows.

(6) Decrypt ($\mathrm{decrypt}(\mathrm{TCT}, \mathrm{SK}) \to m$). The decryption algorithm takes as input the partially decrypted ciphertext TCT and the user's secret key SK. Then, it decrypts the ciphertext as follows:

(1) If there is no attribute revoked, namely, $\mathrm{TCT} = (C, E)$, then the user computes

$$\frac{C}{E^z} = m \cdot \frac{e(g,g)^{\beta s}}{\left(e(g,g)^{\beta s/z}\right)^z} = m. \tag{13}$$

(2) If the attribute x of users list RL_x is revoked, namely, $\mathrm{TCT} = (\mathrm{TCT}', \mathrm{Hdr}_x')$, then we have the following.

Here, we have $\mathrm{TCT}' = (C', \{B_i\}_{\rho(i)\neq x}, C_{x,1}', C_{x,2}', D)$ and $\mathrm{Hdr}_x' = (\widetilde{C}, E_x, F_x)$, and then the user computes

$$\begin{aligned}
\widetilde{C} \cdot \frac{F_x}{(E_x)^z} &= v_x \cdot e\left(g_n, g_1\right)^{\tilde{s}} \cdot e\left(g_{\mathrm{ID}}, v\right)^{\tilde{s}} \\
&\quad \cdot \frac{e\left(g_{n+1}, g\right)^{-\tilde{s}}}{\left(e(g,g)^{\alpha^{\mathrm{ID}} \gamma \tilde{s}/z}\right)^z} = v_x.
\end{aligned} \tag{14}$$

If the attributes set S satisfies the access control policy $(\mathbf{M}, \rho)$, then the CSP computes the values $\{w_i \in \mathbb{Z}_p\}_{i\in I}$ satisfying $\sum_{i\in I} w_i \mathbf{M}_i = (1, 0, \ldots, 0)$ in polynomial time and continues to compute

$$\begin{aligned}
B_x &= e\left(C_{x,1}', L\right) e\left(C_{x,2}', \left(K_{\rho(x)}\right)^{v_x}\right) \\
&= e\left(g_1^{\lambda_x} h_{\rho(x)}^{-r_x}, g^r\right) \cdot e\left(\left(g^{r_x}\right)^{1/v_x}, \left(h_{\rho(x)}^r\right)^{v_x}\right) \\
&= e\left(g_1^{\lambda_x}, g^r\right) = e(g,g)^{\alpha r \lambda_x}, \\
B &= \prod_{i\in I} (B_i)^{w_i} = \prod_{i\in I} \left(e(g,g)^{\alpha r \lambda_i}\right)^{w_i} = e(g,g)^{\alpha r s}, \\
E &= \frac{D}{B} = \frac{e(g,g)^{\beta s/z} e(g,g)^{\alpha r s}}{e(g,g)^{\alpha r s}} = e(g,g)^{\beta s/z}, \\
\frac{C}{E^z} &= m \cdot \frac{e(g,g)^{\beta s}}{\left(e(g,g)^{\beta s/z}\right)^z} = m.
\end{aligned} \tag{15}$$

3.4. Security Proof

Theorem 4. *If the decisional q-Parallel BDHE assumption holds in $\mathbb{G}$ and $\mathbb{G}_T$, then there exists no polynomial time attacker to break our proposed CP-ABE scheme with attribute level user revocation selectively, where the challenge matrix is $\mathbf{M}^*(l^* \times n^*)$ with $l^*, n^* \leq q$.*

Proof. If there exists an attacker $\mathcal{A}$ who can selectively break our proposed CP-ABE scheme with a nonnegligible advantage $\varepsilon = \mathrm{Adv}_{\mathcal{A}}$, where the challenge matrix is $\mathbf{M}^*(l^* \times n^*)$ with

$l^*, n^* \le q$, then we can construct a challenger $\mathcal{B}$ to break the decisional q-Parallel BDHE assumption successfully. □

Init. The challenger $\mathcal{B}$ takes as input a q-Parallel BDHE challenge $\vec{y}, T$. In addition, the attacker $\mathcal{A}$ gives the challenge access control policy $(\mathbf{M}^*, \rho^*)$ and the revocation users list RL_{x^*} of attribute x^* where the matrix $\mathbf{M}^*$ has n^* columns.

Setup. The challenger $\mathcal{B}$ chooses a random exponent $\beta' \in \mathbb{Z}_p$ and computes $e(g,g)^{\beta} = e(g,g)^{\beta'} \cdot e(g^{\alpha}, g^{\alpha^q})$, where it implicitly sets $\beta = \beta' + \alpha^{q+1}$. In addition, it sets the broadcast users set as

$$\begin{aligned} \widehat{N} &= \mathrm{RL}_{x^*} \cap \{1, 2, \ldots, n\}, \\ N &= \{1, 2, \ldots, n\} \setminus \widehat{N}. \end{aligned} \tag{16}$$

Then, $\mathcal{B}$ selects a random exponent $u \in \mathbb{Z}_p$ and sets $v = g^u \prod_{k \in N} g_{q+1-k}$.

Next, $\mathcal{B}$ sets the group parameters $h_1, h_2, \ldots, h_U$, and for each x $(1 \le x \le U)$, $\mathcal{B}$ selects a random exponent $z_x \in \mathbb{Z}_p$. Let X denote the set of i satisfying $\rho^*(i) = x$; then, h_x is set as

$$h_x = g^{z_x} \prod_{i \in X} g^{a\mathbf{M}^*_{i,1}/b_i} \cdot g^{a^2 \mathbf{M}^*_{i,2}/b_i} \cdots g^{a^{n^*} \mathbf{M}^*_{i,n^*}/b_i}. \tag{17}$$

Note that if $X = \varnothing$, then we have $h_x = g^{z_x}$. In addition, we can say that h_x is distributed randomly because of the randomness of z_x.

Finally, $\mathcal{B}$ sends to $\mathcal{A}$ the public key PK as

$$\mathrm{PK} = \left(g, g_1, \ldots, g_q, g_{q+2}, \ldots, g_{2q}, v, e(g,g)^{\beta}, h_1, \ldots, h_U\right). \tag{18}$$

Query Phase 1. $\mathcal{A}$ makes to $\mathcal{B}$ a series of queries including the key generation query $\mathcal{O}_{\mathrm{kg}}$ and the ciphertext reencryption query $\mathcal{O}_{\mathrm{ree}}$.

(i) $\mathcal{A}$ makes to $\mathcal{B}$ a key generation query $\mathcal{O}_{\mathrm{kg}}$ associated with the identity ID_j and the attributes set S_j; if $\mathrm{ID}_j \notin \mathrm{RL}_{x^*}$, then we set the attributes set $S'_j = S_j$; otherwise, we set $S'_j = S_j \setminus \{x^*\}$. In addition, if S'_j satisfies the challenge access control policy $(\mathbf{M}^*, \rho^*)$, then $\mathcal{B}$ outputs $\perp$; otherwise, it generates the secret key as follows.

$\mathcal{B}$ first computes the vector $\vec{\mathbf{w}} = (w_1, \ldots, w_{n^*}) \in \mathbb{Z}_p^n$, where $w_1 = -1$, and for all $\rho^*(i) \in S'_j$, it satisfies $\mathbf{M}^*_i \vec{\mathbf{w}}^T = 0$. Note that the vector can be found in polynomial time according to the definition of LSSS.

Then, $\mathcal{B}$ chooses a random parameter $t \in \mathbb{Z}_p$ and defines the exponent r as

$$r = t + w_1 \alpha^q + w_2 \alpha^{q-1} + \cdots + w_{n^*} \alpha. \tag{19}$$

Next, $\mathcal{B}$ computes the key component L' as

$$L' = g^t \cdot \prod_{i=1,\ldots,n^*} \left(g^{\alpha^{q+1-i}}\right)^{w_i} = g^r. \tag{20}$$

According to the definition of r and $w_1 = -1$, we know that $g^{\alpha r}$ includes the item $g^{-\alpha^{q+1}}$. Although $g^{-\alpha^{q+1}}$ is not given in the assumption, it can be canceled by multiplying $g^{\alpha r}$ with $g^{\beta} = g^{\beta'} g^{\alpha^{q+1}}$, because we implicitly set $\beta = \beta' + \alpha^{q+1}$ when generating the key component $\widetilde{K}'$. In detail, it is constructed as follows:

$$\begin{aligned} \widetilde{K}' &= g^{\beta'} g^{\alpha^{q+1}} g^{\alpha t} g^{-\alpha^{q+1}} \prod_{i=2,\ldots,n^*} \left(g^{\alpha^{q+2-i}}\right)^{w_i} \\ &= g^{\beta'} g^{\alpha t} \prod_{i=2,\ldots,n^*} \left(g^{\alpha^{q+2-i}}\right)^{w_i}. \end{aligned} \tag{21}$$

Then, $\mathcal{B}$ will compute the key component K'_i, $\forall i \in S'_j$. For each attribute $i \in S'_j$, if there exists no row k satisfying $\rho^*(k) = i$, then we set $K'_i = (L')^{z_i}$; otherwise, let X denote the set of all the rows k satisfying $\rho^*(k) = i$, and then we set K'_i as

$$\begin{aligned} K'_i = \left(L'\right)^{z_i} \prod_{i \in X} \prod_{j=1,\ldots,n^*} \Bigg(& g^{(\alpha^j/b_i)t} \\ & \cdot \prod_{\substack{k=1,\ldots,n^*, \\ k \ne j}} \left(g^{(\alpha^{q+1+j-k}/b_i) w_k}\right) \Bigg)^{\mathbf{M}^*_{i,j}}. \end{aligned} \tag{22}$$

Next, $\mathcal{B}$ will set the key component K' for the user $\mathrm{ID}_j \notin \mathrm{RL}_{x^*}$. Similarly, $g^{\alpha r}$ includes the item $g^{-\alpha^{q+1}}$ that is not given in the assumption. However, we set the value v as $v = g^u \prod_{k \in N} g_{q+1-k}$ and we have $g^{\alpha^{\mathrm{ID}_j} \gamma} = (g^u \prod_{k \in \widehat{N}} g_{q+1-k})^{\alpha^{\mathrm{ID}_j}}$. Moreover, because $\mathrm{ID}_j \notin \mathrm{RL}_{x^*}$, namely, $\mathrm{ID}_j \in N$, $g^{\alpha^{\mathrm{ID}} \gamma}$ includes the term $g^{\alpha^{q+1}}$ that can be canceled by the term $g^{-\alpha^{q+1}}$ included in $g^{\alpha r}$:

$$\begin{aligned} K' &= g^{\alpha^{\mathrm{ID}} \gamma} g^{\alpha r} \\ &= \left(g^u \prod_{k \in N} g_{q+1-k}\right)^{\alpha^{\mathrm{ID}}} \cdot g^{\alpha t} g^{-\alpha^{q+1}} \prod_{i=2,\ldots,n^*} \left(g^{\alpha^{q+2-i}}\right)^{w_i} \\ &= \left(g^{\alpha^{\mathrm{ID}}}\right)^u \left(\prod_{k \in N \setminus \{\mathrm{ID}_j\}} g_{q+1-k+\mathrm{ID}_j}\right) \cdot g_{q+1-\mathrm{ID}_j+\mathrm{ID}_j} \\ &\quad \cdot g^{\alpha t} g^{-\alpha^{q+1}} \prod_{i=2,\ldots,n^*} \left(g^{\alpha^{q+2-i}}\right)^{w_i} \\ &= \left(g^{\alpha^{\mathrm{ID}}}\right)^u \left(\prod_{k \in N \setminus \{\mathrm{ID}_j\}} g_{q+1-k+\mathrm{ID}_j}\right) \\ &\quad \cdot g^{\alpha t} \prod_{i=2,\ldots,n^*} \left(g^{\alpha^{q+2-i}}\right)^{w_i}. \end{aligned} \tag{23}$$

Once the key components are all generated, the challenger $\mathcal{B}$ will select a random exponent $z \in \mathbb{Z}_p^*$ and set the outsourced transformation key TK as

$$\begin{aligned}\mathrm{TK} = \Big(& K = \left(K'\right)^{1/z},\ \widetilde{K} = \left(\widetilde{K}'\right)^{1/z},\ L \\ & = \left(L'\right)^{1/z},\ \{K_i\}_{i\in S_j'} = \left\{\left(K_i'\right)^{1/z}\right\}_{i\in S_j'}\Big).\end{aligned} \tag{24}$$

Therefore, the secret key is set as SK = (z, TK). Finally, $\mathcal{B}$ sends the transformation key TK to the attacker $\mathcal{A}$.

(ii) $\mathcal{A}$ makes to $\mathcal{B}$ a ciphertext reencryption query $\mathcal{O}_{\mathrm{ree}}$ associated with the revocation users list RL_x of attribute x and the ciphertext $\mathrm{CT} = (C, C_0, \{C_{i,1}, C_{i,2}\}_{i=1}^{l})$. Then, $\mathcal{B}$ generates the reencrypted ciphertext as follows.

$\mathcal{B}$ first selects a random exponent $v_x \in \mathbb{Z}_p^*$ and computes

$$\begin{aligned}\mathrm{CT}' = \Big\{ & C' = C,\ C_0' = C_0,\ \rho(i) \neq x\colon C_{i,1}' = C_{i,1},\ C_{i,2}' \\ & = C_{i,2},\ \rho\left(i = x\colon C_{i,1}' = C_{i,1},\ C_{i,2}' = \left(C_{i,2}\right)^{1/v_x}\right)\Big\}.\end{aligned} \tag{25}$$

Next, $\mathcal{B}$ selects random parameters $\widetilde{s}, \widetilde{v}_2, \ldots, \widetilde{v}_n \in \mathbb{Z}_p$ and defines the vector $\widetilde{\mathbf{v}} = (\widetilde{s}, \widetilde{v}_2, \ldots, \widetilde{v}_n)$. For each row $\mathbf{M}_i$ of the matrix $\mathbf{M}$, $\mathcal{B}$ computes the inner product $\widetilde{\lambda}_i = \mathbf{M}_i \cdot \widetilde{\mathbf{v}}$. Then, $\mathcal{B}$ selects a random exponent $\widetilde{r}_i \in \mathbb{Z}_p$ and defines the broadcast users set as $N = q \setminus \{\mathrm{RL}_x\}$. Finally, it encrypts the exponent v_x to generate the ciphertext header as follows:

$$\begin{aligned}\mathrm{Hdr}_{x^*} = \Big(& \mathrm{RL}_{x^*},\ \widetilde{C} = v_x \cdot e\left(g_n, g_1\right)^{\widetilde{s}},\ \widetilde{C}_0 = g^{\widetilde{s}},\ \widetilde{C}_1 \\ & = \left(g^u\right)^{\widetilde{s}},\ \left\{\widetilde{C}_{i,1} = \left(g_1\right)^{\widetilde{\lambda}_i} h_{\rho(i)}^{-\widetilde{r}_i},\ \widetilde{C}_{i,2} = g^{\widetilde{r}_i}\right\}_{i=1}^{l}\Big).\end{aligned} \tag{26}$$

Note that $\widetilde{C}_1$ is a correctly distributed ciphertext component which is demonstrated as follows:

$$\begin{aligned}\widetilde{C}_1 = \left(g^u\right)^{\widetilde{s}} &= \left(g^u \prod_{k\in N} g_{q+1-k} \cdot \left(\prod_{j\in N} g_{q+1-k}\right)^{-1}\right)^{\widetilde{s}} \\ &= \left(v\left(\prod_{j\in N} g_{q+1-k}\right)^{-1}\right)^{\widetilde{s}}.\end{aligned} \tag{27}$$

Therefore, the final reencrypted ciphertext is set as $\mathrm{CT}'' = (\mathrm{CT}', \mathrm{Hdr}_x)$.

Challenge. The attacker $\mathcal{A}$ submits to the challenger $\mathcal{B}$ two messages m_0 and m_1 with the equal length. Then, $\mathcal{B}$ selects a random coin $\beta \in \{0, 1\}$ and generates the challenge ciphertext components as

$$\begin{aligned}C^* &= m_\beta \cdot T \cdot e\left(g^s, g^{\beta'}\right), \\ C_0^* &= g^s.\end{aligned} \tag{28}$$

Next, $\mathcal{B}$ selects random parameters $y_2', \ldots, y_{n^*}' \in \mathbb{Z}_p$ and then sets the vector $\vec{\mathbf{v}} = (s, sa + y_2', sa^2 + y_3', \ldots, sa^{n-1} + y_{n^*}') \in \mathbb{Z}_p^{n^*}$ to implicitly share the key s. For $i = 1, 2, \ldots, n^*$, $\mathcal{B}$ defines R_i as the set of all $k \neq i$ satisfying $\rho^*(i) = \rho^*(k)$. Finally, $\mathcal{B}$ selects random exponents $r_1', r_2', \ldots, r_l' \in \mathbb{Z}_p$ and sets the challenge ciphertext components $C_{i,1}^*$ and $C_{i,2}^*$ as follows:

$$\begin{aligned}C_{i,1}^* &= g^{-r_i'} g^{-sb_i}, \\ C_{i,2}^* &= h_{\rho^*(i)}^{r'} \left(\prod_{j=2,\ldots,n^*} \left(g^\alpha\right)^{\mathbf{M}_{i,j}^* y_j'}\right) \cdot \left(g^{sb_i}\right)^{-z_{\rho^*(i)}} \\ &\quad \cdot \left(\prod_{k\in R_i} \prod_{j=1,\ldots,n^*} \left(g^{\alpha^j \cdot s \cdot (b_i/b_k)}\right)^{\mathbf{M}_{k,j}^*}\right).\end{aligned} \tag{29}$$

Query Phase 2. $\mathcal{A}$ continues to make to $\mathcal{B}$ a series of queries including the key generation query $\mathcal{O}_{\mathrm{kg}}$ and the ciphertext reencryption query $\mathcal{O}_{\mathrm{ree}}$ as in *Query Phase 1*.

Guess. The attacker $\mathcal{A}$ outputs its guess β' for β. If $\beta = \beta'$, then $\mathcal{A}$ outputs 0 denoting $T = e(g, g)^{\alpha^{q+1} s}$; otherwise, it outputs 1 denoting T is a random parameter in $\mathbb{G}_T$.

If $T = e(g, g)^{\alpha^{q+1} s}$, then $\mathcal{B}$ plays the proper security game, so we have

$$\Pr\left[\mathcal{B}\left(\vec{y},\ T = e\left(g, g\right)^{\alpha^{q+1} s}\right) = 0\right] = \frac{1}{2} + \mathrm{Adv}_{\mathcal{A}}. \tag{30}$$

Otherwise, T is a random element in $\mathbb{G}_T$; namely, m_β is completely random in the view of $\mathcal{A}$, so we have

$$\Pr\left[\mathcal{B}\left(\vec{y},\ T = R\right) = 0\right] = \frac{1}{2}. \tag{31}$$

4. Analysis

In this part, we will compare our proposed CP-ABE scheme with several existing revocation schemes in terms of functionality, storage cost, communication cost, and computation efficiency. The notations that will be used are described as follows: $|C_1|$ denotes the bit size of an element in $\mathbb{G}$; $|C_T|$ denotes the bit size of an element in $\mathbb{G}_T$; $|C_p|$ denotes the bit size of an element in $\mathbb{Z}_p^*$; $C_{\mathcal{T}}$ denotes the size of access control matrix associated with the ciphertext; $|C_k|$ denotes the bit size of the key encryption key in Hur's scheme [13]; t denotes the number of attributes associated with the ciphertext; k denotes the number of attributes associated with the secret key of a user; n_a denotes the number of all attributes in the system; n_u denotes the number of all users in the system.

4.1. Functionality. The functionality comparison is demonstrated in Table 1, from which we can see that Liang's scheme achieve the system level user revocation; namely, once an attribute of some user is revoked, he will lose all the access permissions in the system, which is impractical in the normal application. However, our scheme, Hur's scheme, and

Table 1: Comparison of functionalities.

Scheme	Access control granularity	Model	Assumption
Liang	System level user revocation	Standard	DBDH
Hur	Attribute level user revocation	Generic group	—
Yang	Attribute level user revocation	Random oracle	q-Parallel BDHE
Ours	Attribute level user revocation	Standard	q-Parallel BDHE

Yang's scheme achieve the attribute level user revocation; namely, the revocation of some attribute has no effect on the access permissions of other legitimate attributes. In addition, compared with the generic group model of Hur's scheme and the random oracle model of Yang's scheme, only our scheme is provably secure based on q-Parallel BDHE assumption in the standard model, which has stronger security.

4.2. Storage Cost . The storage cost comparison is demonstrated in Table 2. The storage cost of attribute authority (AA) is mainly generated by the master key MK. Our scheme and Hur's scheme have short and constant master key; however, the master key in Liang's scheme grows linearly with the number n_u of all users in the system and in Yang's scheme grows linearly with the number n_a of all attributes in the system. The storage cost of data owner (DO) is mainly generated by the public key PK. Hur's scheme has the shortest public key which is constant. The public key in Yang's scheme grows linearly with the number n_a of all attributes in the system and in Liang's scheme grows linearly with the number n_a of all attributes and the column vector $C_{\mathcal{T}}/t$ of access control matrix with each other as the slope and in our scheme grows linearly with the number n_a of all attributes and the number n_u of all users, however, with constant slope compared with Liang's scheme. The storage cost of cloud service provider (CSP) is mainly generated by the ciphertext and ciphertext header. Liang's scheme only achieves user revocation in which the key updating is implemented by using the method of subset cover and the ciphertext needs not to be updated; therefore, the ciphertext grows linearly with the size $C_{\mathcal{T}}$ of the access control matrix. Yang's scheme updates the key through the interaction between the AA and the data user (DU) and also updates the corresponding ciphertext associated with the revoked attribute; therefore, the ciphertext grows linearly with the number t of attributes associated with the ciphertext. In Hur's scheme, once the DO sends the ciphertext to the CSP, the CSP generates the corresponding ciphertext header for each attribute group. Therefore, the storage cost includes the ciphertext and ciphertext header; moreover, the ciphertext grows linearly with the number t of attributes associated with the ciphertext, and the ciphertext header grows linearly with the number t of attributes and the number n_u of all users in the system with each other as the slope. In our scheme, if some attribute is revoked, then the CSP selects a new exponent to update the ciphertext corresponding to the revoked attribute and then encrypts the exponent to generate the corresponding ciphertext header. Therefore, the storage cost also includes the ciphertext and ciphertext header; moreover, the ciphertext and ciphertext header both grow linearly with the number t of attributes associated with the ciphertext. The storage cost of the DU is mainly generated by the secret key. Our scheme and Yang's scheme have shorter secret key which grows linearly with the number k of attributes associated with the secret key. In Liang's scheme, the secret key is generated by using a binary tree; therefore, the size of secret key is associated with the number k of attributes, the column vector $C_{\mathcal{T}}/t$ of access control matrix, and the number n_u of all users in the system. In addition, in Liang's scheme, the key updating is implemented by using the method of subset cover, so the storage cost also includes the updating key that grows linearly with the smallest cover set. In Hur's scheme, every user needs to store a plenty of key encryption keys to decrypt the corresponding exponents for key updating; therefore, the size of secret key not only grows linearly with the number k of attributes but only grows logarithmically with the number n_u of all users in the system.

4.3. Communication Cost. The communication cost comparison is demonstrated in Table 3. The communication cost is mainly generated by the key and the ciphertext. The communication cost between the attribute authority (AA) and the data user (DU) is mainly generated by the secret key of user. In Liang's scheme, for every revocation, the AA needs to generate a new updating key which then is sent to the DU; therefore, it causes $2(n_u - n_m)\log(n_u/(n_u - n_m))|C_1|$ size communication cost additionally. In Yang's scheme, for every revocation, the AA needs to communicate with the DU for updating the key; therefore, it causes $2|C_1|$ size communication cost additionally between the AA and DU. In addition, the communication cost between the AA and data owner (DO) is mainly generated by the public key, and in Yang's scheme, the AA needs to update the public key for every attribute revocation; therefore, it generates $2|C_1|$ size communication cost also. The communication cost between the cloud service provider (CSP) and the DU is generated by the ciphertext, and in Hur's scheme, the CSP needs not only to send the ciphertext but also to generate the key encryption keys, which causes $(\log n_u + 1)|C_k|$ size communication cost; in addition, it also needs to send $((t \cdot n_u)/2)|C_p|$ size ciphertext header. In our proposed CP-ABE scheme, for every revoked attribute, the CSP selects a new exponent to implement the ciphertext updating and then encrypts the exponent to generate the ciphertext header, which causes $(2t + 2)|C_1| + |C_T|$ size communication size additionally. However, because we outsource the decryption to the CSP, the DU needs to send $(k + 3)|C_1|$ size transformation key to the CSP for partial decryption. If there is no attribute revoked, then the CSP generates only two elements in $\mathbb{G}_T$; otherwise, the CSP generates $t + 1$ elements in $\mathbb{G}_T$ and two elements in $\mathbb{G}$ corresponding to the ciphertext and three elements in $\mathbb{G}_T$ corresponding to the ciphertext header. In addition, the communication cost between the CSP and the DO is mainly generated by the ciphertext.

Table 2: Comparison of storage costs.

Entity	Liang	Hur	Yang	Ours
AA	$\lvert C_1\rvert + \left(2^{(\log n_u+1)} + 1\right)\lvert C_p\rvert$	$\lvert C_p\rvert + \lvert C_1\rvert$	$(4 + n_a)\lvert C_p\rvert$	$2\lvert C_p\rvert + \lvert C_1\rvert$
DO	$((C_{\mathcal{T}}/t)\cdot n_a + 6)\lvert C_1\rvert + \lvert C_T\rvert + \lvert C_p\rvert$	$2\lvert C_1\rvert + \lvert C_T\rvert$	$(2n_a + 4)\lvert C_1\rvert + \lvert C_T\rvert$	$(n_a + 2n_u + 1)\lvert C_1\rvert + \lvert C_T\rvert$
CSP	$(C_{\mathcal{T}} + 3)\lvert C_1\rvert + \lvert C_T\rvert$	$(2t + 1)\lvert C_1\rvert + \lvert C_T\rvert + ((t\cdot n_u)/2)\lvert C_p\rvert$	$(3t + 1)\lvert C_1\rvert + \lvert C_T\rvert$	$(4t + 3)\lvert C_1\rvert + 2\lvert C_T\rvert$
DU	$(k + 3 + C_{\mathcal{T}}/t)(\log n_u + 1)\lvert C_1\rvert + 2(n_u - n_m)\log(n_u/(n_u - n_m))\lvert C_1\rvert$	$(2k + 1)\lvert C_1\rvert + (\log n_u + 1)C_k$	$(k + 2)\lvert C_1\rvert$	$(k + 3)\lvert C_1\rvert + \lvert C_p\rvert$

Table 3: Comparison of communication costs.

Entity	Liang	Hur	Yang	Ours
AA & DU	$(k + 3 + C_{\mathcal{T}}/t)(\log n_u + 1)\lvert C_1\rvert + 2(n_u - n_m)\log(n_u/(n_u - n_m))\lvert C_1\rvert$	$(2k + 1)\lvert C_1\rvert$	$(k + 4)\lvert C_1\rvert$	$(k + 3)\lvert C_1\rvert + \lvert C_p\rvert$
AA & DO	$((C_{\mathcal{T}}/t)\cdot n_a + 6)\lvert C_1\rvert + \lvert C_T\rvert + \lvert C_p\rvert$	$2\lvert C_1\rvert + \lvert C_T\rvert$	$(2n_a + 6)\lvert C_1\rvert + \lvert C_T\rvert$	$(n_a + 2n_u + 1)\lvert C_1\rvert + \lvert C_T\rvert$
CSP & DU	$(C_{\mathcal{T}} + 3)\lvert C_1\rvert + \lvert C_T\rvert$	$(2t + 1)\lvert C_1\rvert + \lvert C_T\rvert + ((t\cdot n_u)/2)\lvert C_p\rvert + (\log n_u + 1)\lvert C_k\rvert$	$(3t + 1)\lvert C_1\rvert + \lvert C_T\rvert$	$(k + 3)\lvert C_1\rvert + 2\lvert C_T\rvert$ or $(k + 5)\lvert C_1\rvert + (t + 4)\lvert C_T\rvert$
CSP & DO	$(C_{\mathcal{T}} + 3)\lvert C_1\rvert + \lvert C_T\rvert$	$(2t + 1)\lvert C_1\rvert + \lvert C_T\rvert$	$(3t + 1)\lvert C_1\rvert + \lvert C_T\rvert$	$(2t + 1)\lvert C_1\rvert + \lvert C_T\rvert$

4.4. Computation Efficiency. In order to evaluate the computation efficiency of our proposed CP-ABE scheme with attribute level user revocation, we implement our scheme on a 3.4 GHZ processor PC with 64-bit Ubuntu 14.04 operating system, Intel® Core™ i7-3770CPU and 4 G memory. The public key is selected to provide a 128-bit security level. In addition, the experiment uses a 160-bit elliptic curve group based on the pairing-based cryptography library (PBC-0.5.14) [20] and cpabe-0.11 [21] which selects the supersingular curve $y^2 = x^3 + x$ over 512-bit finite field. The experimental data are obtained by computing the average value for 20 times. In this experiment, the time of PBC library computing a pairing operation is approximately 5.3 ms, and the time of computing an exponent operation in $\mathbb{G}$ and $\mathbb{G}_T$ is approximately 6.2 ms and 0.6 ms, respectively. In addition, the selection time of a random element in $\mathbb{G}$ and $\mathbb{G}_T$ is approximately 14 ms and 1.4 ms, respectively, by using the operation/dev/urandom in Ubuntu 14.04 operating system.

In this paper, we compare our scheme with several related schemes in terms of key generation time, encryption time, decryption time, and reencryption time; moreover, we set $C_{\mathcal{T}}/t = 6$, $n_u = 8$.

From Figure 2, we can see that the key generation time grows linearly with the number of attributes, and our key generation time is slightly higher than that of Yang's scheme; however, it is better than that of Hur's scheme and Liang's scheme. In particular, the key generation time in Liang's scheme is associated with not only the number of attributes but also the column vector $C_{\mathcal{T}}/t$ of access control matrix and the number n_u of all users in the system; therefore, its key generation time is much larger than the other three schemes.

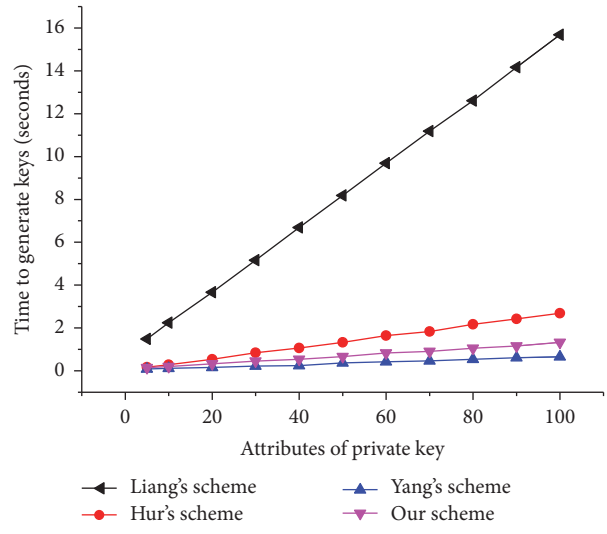


Figure 2: Key generation time.

From Figure 3, we can see that the encryption time grows linearly with the number of attributes associated with the access control policy. Our encryption time is slightly higher than that of Hur's scheme and, however, is better than that of Yang's scheme and Liang's scheme. Note that the encryption in Hur's scheme involves some polynomial operations; however, the running time is very short which is omitted here. The encryption time in Liang's scheme is not only associated with the number of attributes corresponding to the access control policy but also associated with the column vector $C_{\mathcal{T}}/t$ of access control matrix; therefore,

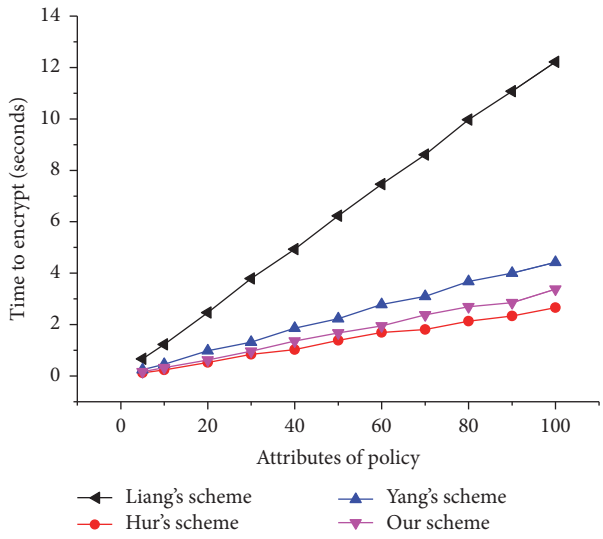


Figure 3: Encryption time.

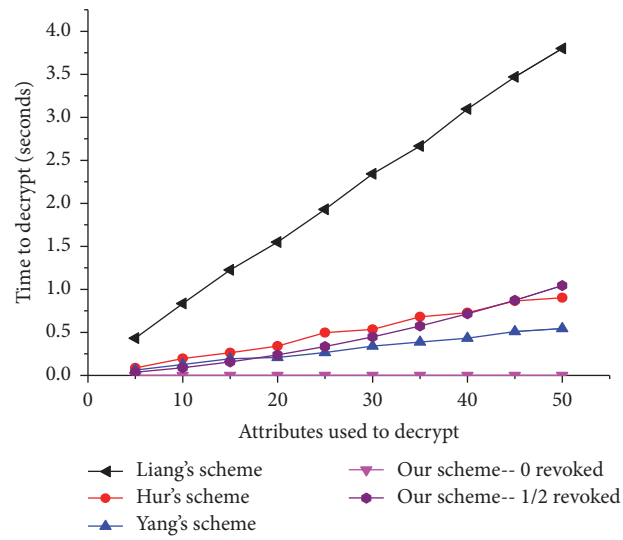


Figure 4: Decryption time.

the encryption time is much larger than the other three schemes.

In the decryption experiment, the computation time is mainly influenced by the number of attributes used in decryption. In order to demonstrate the experimental results better, we suppose that all the intermediate nodes in the binary tree use the (n, n)-threshold gates. In addition, our scheme is demonstrated under two circumstances; namely, no attribute is revoked and 50% attributes are revoked. From Figure 4, we can see that the decryption time in our scheme with 50% attributes revoked, Liang's scheme, Hur's scheme, and Yang's scheme grows linearly with the number of attributes used in decryption. Moreover, our scheme with no attribute revoked uses outsourced decryption, so the user needs only one exponent operation in $\mathbb{G}_T$. In addition, the decryption time of our scheme with 50% attributes revoked is a quadratic function for the attributes used in decryption; however, we also uses outsourced decryption which decreases the decryption time of user greatly. From Figure 4, we can see that when the number of attributes used in decryption locates in a certain range, the decryption time of our scheme with 50% attributes revoked is smaller than the other three schemes, and as the number of attributes used to decrypt increases, the decryption time goes over Yang's scheme and Hur's scheme successively, however, within acceptable range.

In addition, the comparison of reencryption times is shown in Figure 5. If there exists some attribute to be revoked, then the key or the ciphertext should be updated. Yang's scheme and Liang's scheme mainly implement the key updating while Hur's scheme and our scheme mainly implement the ciphertext updating. Therefore, from Figure 5, we can see that the reencryption time in Hur's scheme and our scheme is larger and grows linearly with the number of attributes associated with access control policy. However, all these computations are implemented by the CSP that has a plenty of computing resources. Although the reencryption time in Yang's scheme and Liang's scheme is shorter, it

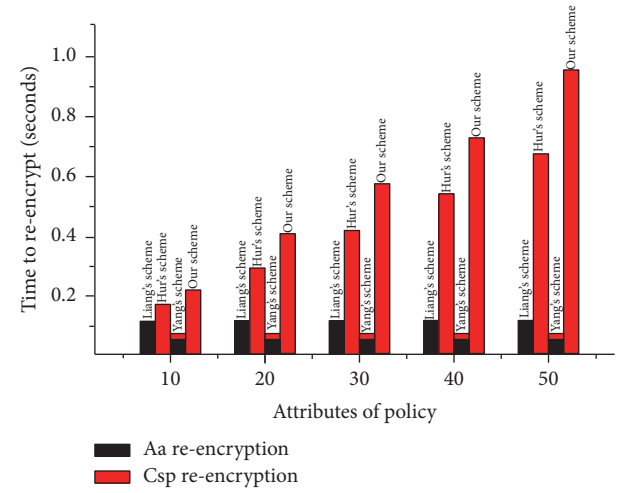


Figure 5: Reencryption time.

requires AA to implement the key updating. As we all know, the computation resources of AA are limited, which may be the bottleneck in the system.

5. Conclusion

In this paper, we propose a CP-ABE scheme which can achieve the attribute level user revocation. In this scheme, if some attribute of a user is revoked, then the ciphertext corresponding to the revoked attribute is updated so that only the user, whose attributes set satisfies the access control policy and has not been revoked, can carry out the key updating to decrypt the ciphertext successfully. The security of our scheme is proved secure based on the q-Parallel BDHE assumption in the standard model. Finally, the performance analysis and experimental verification are carried out, and the experimental results show that although our scheme

increases the computation cost of the CSP in order to achieve the attribute revocation, it does not require the participation of the AA, which decreases the computation cost of the AA. Moreover, the user does not need to store additional parameters to carry out the attribute revocation; thus, it greatly saves the storage space.

Conflicts of Interest

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

Acknowledgments

The authors acknowledge the important comments given by the instructors and colleagues. This study acquired support from National Key Research Program of China "Collaborative Precision Position Project" (Grant no. 2016YFB0501900).

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based Encryption," in *Advances in cryptology—EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Computer Sci.*, pp. 457–473, Springer, Berlin, Germany, 2005.

[2] U. C. Yadav, "Ciphertext-policy attribute-based encryption with hiding access structure," in *Proceedings of the 2015 5th IEEE International Advance Computing Conference, (IACC '15)*, pp. 6–10, India, June 2015.

[3] T. Naruse, M. Mohri, and Y. Shiraishi, "Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating," *Human-centric Computing and Information Sciences*, vol. 5, no. 1, pp. 1–13, 2015.

[4] H. Wang, B. Yang, and Y. Wang, "Server aided ciphertext-policy attribute-based encryption," in *proceedings of the IEEE International Conference on Advanced Information Networking Applications Workshops*, pp. 440–444, Gwangju, Korea, 2015.

[5] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Large universe decentralized key-policy attribute-based encryption," *Security and Communication Networks*, vol. 8, no. 3, pp. 501–509, 2015.

[6] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, "Performance evaluation of Attribute-Based Encryption: toward data privacy in the IoT," in *proceedings of the 2014 1st IEEE International Conference on Communications (ICC '14)*, pp. 725–730, Sydney, Australia, June 2014.

[7] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 195–203, November 2007.

[8] J. Staddon, P. Golle, M. Gagne, and P. Rasmussen, "A content-driven access control system," in *Proceedings of the 7th Symposium on Identity and Trust on the Internet (IDtrust '08)*, pp. 26–35, Gaithersburg, Maryland, USA, March 2008.

[9] X. Liang, R. Lu, and X. Lin, "Ciphertext policy attribute based encryption with efficient revocation," in *Proceedings of the IEEE Symposium on Security Privacy*, vol. 2008, pp. 321–334, 2010.

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

[11] A. Boldyreva, V. Goyal, and V. Kumart, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM conference on Computer and Communications Security (CCS '08)*, pp. 417–426, Alexandria, VA, USA, October 2008.

[12] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 99–112, Alexandria, Va, USA, October-November 2006.

[13] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[14] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (ASIACCS '13)*, pp. 523–528, May 2013.

[15] E. Zavattoni, L. J. Perez, S. Mitsunari et al., "Software implementation of an attribute-based encryption scheme," *IEEE Transactions on Computers*, vol. 64, no. 5, pp. 1429–1441, 2015.

[16] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," *Lecture Notes in Computer Science*, vol. 2008, pp. 321–334, 2011.

[17] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pp. 456–465, NY, USA, November 2007.

[18] S. S. Tu, S. Z. Niu, and H. Li, "A fine-grained access control and revocation scheme on clouds," *Concurrency & Computation Practice & Experience*, vol. 28, no. 6, 2012.

[19] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in cryptology—EUROCRYPT 2010*, vol. 6110 of *Lecture Notes in Comput. Sci.*, pp. 62–91, Springer, Berlin, Germany, 2010.

[20] B. Lynn, "The pairing-based cryptography (PBC) library[OL]," 2006, http://crypto.stanford.edu/pbc.

[21] J. Bethencourt, A. Sahai, and B. Waters, "Advanced crypto software collection: the cpabetoolkit[OL]," 2001, http://acsc.cs.utexas.edu/cpabe.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8439706, 9 pages
https://doi.org/10.1155/2017/8439706

Research Article

A Universal High-Performance Correlation Analysis Detection Model and Algorithm for Network Intrusion Detection System

Hongliang Zhu,[1] Wenhan Liu,[1] Maohua Sun,[2] and Yang Xin[1]

[1]*Beijing University of Posts and Telecommunications, Beijing, China*
[2]*Information School, Capital University of Economics and Business, Beijing, China*

Correspondence should be addressed to Hongliang Zhu; zhuhongliang@bupt.edu.cn

Received 2 February 2017; Accepted 3 May 2017; Published 23 May 2017

Academic Editor: Zonghua Zhang

In big data era, the single detection techniques have already not met the demand of complex network attacks and advanced persistent threats, but there is no uniform standard to make different correlation analysis detection be performed efficiently and accurately. In this paper, we put forward a universal correlation analysis detection model and algorithm by introducing state transition diagram. Based on analyzing and comparing the current correlation detection modes, we formalize the correlation patterns and propose a framework according to data packet timing and behavior qualities and then design a new universal algorithm to implement the method. Finally, experiment, which sets up a lightweight intrusion detection system using KDD1999 dataset, shows that the correlation detection model and algorithm can improve the performance and guarantee high detection rates.

1. Introduction

(A) Background. Intrusion detection is a kind of technology which recognizes the intrusion by collecting and analyzing the protected system information [1]. The crucial functions are monitoring Internet and computer system, discovering and distinguishing the intrusion behaviors or attempts, and generating intrusion alarm in real time [2]. Intrusion detection can be thought as a binary technology that distinguishes whether the system state is "normal" or "attack" [3]. The requirements of the intrusion detection system are the detection rate, that is, detection accuracy, followed by real time. Only in high detection speed, it can deal with massive data transmitted in Internet in time, get rid of missing information for low speed [4], cause false negatives and false positives, and minimize the losses brought by the intrusion. However, with the diversification in kind and increasing in number of network attack means, there is a key issue of low detection rate for intrusion detection system [5]. In addition, the traditional intrusion detection system detects slowly and consumes large amounts of resources. With the quick development of network speed, it can not process the massive data transmitted in real time, resulting in a large increase of false positives rate and false negatives rate [6]. Also these problems are becoming more and more serious. Detection rate and detection speed have become important indicators of intrusion detection system real-time requirements [7]. How to build a high detection rate and detection speed intrusion system has become the focus of current research. Figure 1 gives an overview of a universal network intrusion detection framework. A key point in this figure is the use of deep analysis modules to process the associated events.

The deep analysis module plays an important role in intrusion detection system. We can see that the data of deep analysis modules come from two parts, one part is the result of detection on the upper layer and the other part is the raw data. Various data packets or events are processed by correlation algorithm, such as a correlation detection of event frequency and correlation detection for multiple parallel events and so on. The performance of correlation detection influences directly the detection rate and detection speed of intrusion detection system. However, the factors which influence the correlation detection result are various, so it is difficult to extract a unified correlation detection algorithm.

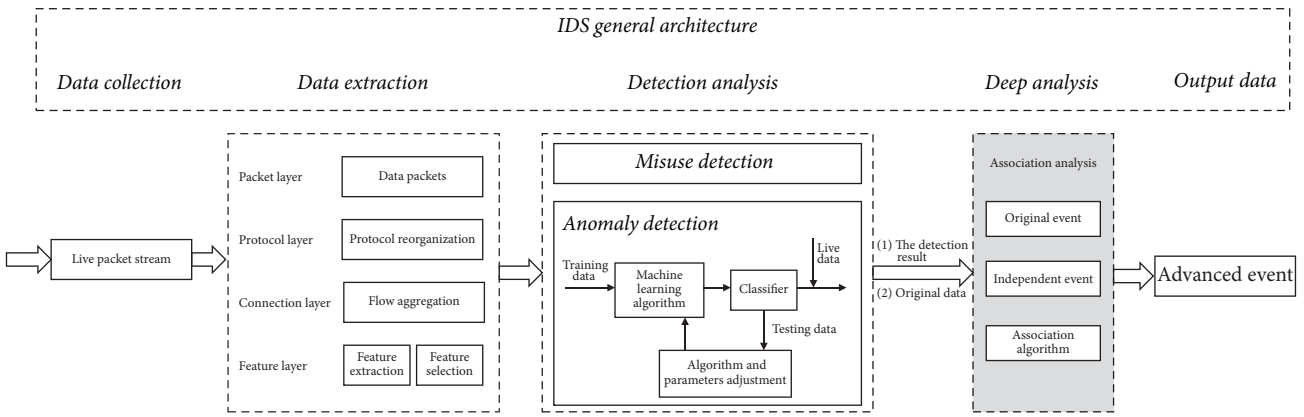


Figure 1: Universal network intrusion detection framework.

Therefore, it is very necessary to find a universal correlation detection algorithm to increase detection rate and speed.

(B) Related Works. In order to detect anomalies in network, correlate parameters from different layers should be combined [8]. Some papers focus on building a new hierarchical framework for intrusion detection as well as data processing based on the feature classification and selection [9–11].

Intrusion detection system has been studied by means of machine learning, and the detection rate has got improvements [12–19]. In addition, intrusion detection has been performed by using feature association technique, and the data set has been used for analysis [20–25].

(C) Contribution. In this paper, we propose a novel method to increase the detection rate of intrusion detection system and improve the detection speed. This method is a correlation analysis detection model based on data packet timing and behavior quality, aiming to solve the problem of versatility, consistency, and the integrity of packet detection. This method enables us to overcome the disadvantage of traditional intrusion detection system.

The rest of this paper is organized as follows. In Section 2, we analyze and compare the current common data packet correlation detection modes briefly. Section 3 presents the generating process of the algorithm in detail. In Section 4, we present the detection process for intrusion detection system and make some experiments. In the end, we conclude the paper in Section 5.

2. System Overview

In intrusion detection system, for single session, there will be false positives when describing threatening events only by single feature, in order to reduce the behavior features of attraction events accurately. However, some papers have pointed out that there is relevance among different attack events [26]. If every session is analyzed separately, we can not identify the attack behavior exactly. While when we consider the related sessions correctly, we can identify an attack event completely. Nowadays, the majority of correlation detecting

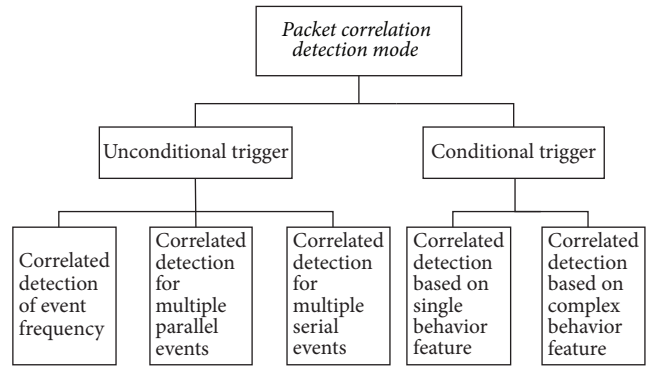


Figure 2: Data packet correlation detection mode.

methods of intrusion detection system are as follows: correlation detection of event frequency, correlation detection for multiple parallel events, correlation detection for multiple serial events, correlation detection based on source IP of the event, correlation detection based on destination IP of the event, correlation detection based on resource of events and destination IP, and correlation of session [27, 28]. There are more and more weaknesses in traditional correlation detection, such as low detection rate and poor accuracy. Thus, we put forward a unified correlation detection algorithm and build a data pack correlation detection model based on the data packet timing and behavior quality, aiming to solve the problem of the versatility, consistency, and integrity of intrusion detection. Figure 2 gives an overview of a data packet correlation detection mode.

According to the behavior features, there are two kinds of intrusion events: one is unconditional trigger and the other is conditional trigger.

(i) Unconditional trigger: correlated detecting based on the order of event occurring, including correlation detection of event frequency, correlation detection for multiple parallel events, and correlation detection for multiple serial events.

(ii) Conditional trigger: correlated detection based on behavior feature, including single behavior feature and complex behavior feature.

3. Correlation Analysis Detection Model

In this section, we present the correlation analysis detection model as follows.

3.1. Concept Definition

Definition 1 (distributed packet flow). The time series is given as $T = \langle t_1, t_2, \ldots, t_i, \ldots \rangle$. The number of nodes is n. A distributed packet flow is defined as $S = \{S_1, S_2, \ldots, S_n\}$; each item of S is S_k $(k = 1, 2, \ldots, n)$, a single data packet, which is the original event collection on T.

Definition 2. The primitive event E is two-tuple (T, A).

(i) T is the timestamp of the original event, that is, the time node of the event on the time series.

(ii) A is the behavioral characteristics of the original event.

Definition 3. The LAMBDA syntax is used to define the different relationships between primitive events: $e_0, e_1, \ldots, e_n$.

(i) Existence: $\bar{e}$ indicates whether the event e_i exists or not.

(ii) Parallel: $e_1 \mid e_2$ indicates that the events e_1 and e_2 are parallel relations.

(iii) Serial: $e_1; e_2$ indicates that the events e_1 and e_2 are serial relations.

Definition 4. The initial state of the system in the state diagram is S_0, the intermediate state is S_i, and N is the termination status. Each node in the following graph represents the current state of the event, and if there is only one transition condition, an arrow arc exists between the two nodes to indicate the transition of the event state. The mark on the arc represents the transition condition.

3.2. Formal Expression of Behavior Detection

3.2.1. Unconditional Trigger Type

(i) A Correlation Detection of Event Frequency. This method detects that the original event that contains the threat behavior feature directly, and then it performs response processing.

The state transition relationship is

$$N = e_i. \tag{1}$$

The state diagram is shown in Figure 3.

(ii) Correlation Detection for Multiple Parallel Events. Some threat behaviors can be detected when multiple events occur at the same time.

Figure 3: The state diagram of a correlation detection of event frequency.

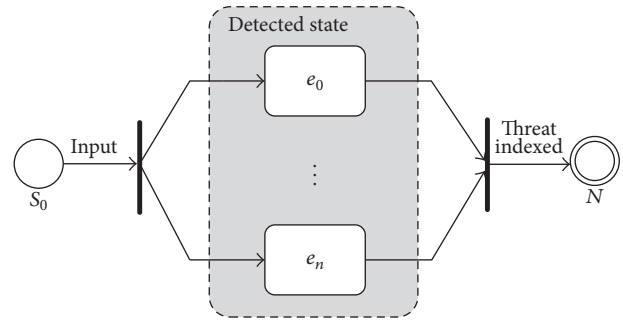


Figure 4: The state diagram of correlation detection for multiple parallel events.

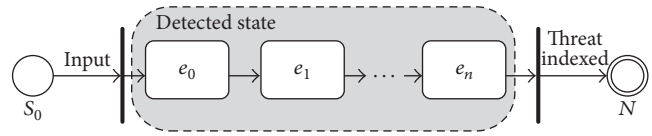


Figure 5: The state transition diagram of correlation detection for multiple serial events.

The state transition relationship is

$$N = e_0 \mid e_1 \mid e_2 \mid \cdots \mid e_n. \tag{2}$$

The state diagram is shown in Figure 4.

(iii) Correlation Detection for Multiple Serial Events. Some threat behaviors can be detected when multiple events occur in sequence.

The state transition relationship is

$$N = e_0; e_1; e_2; \ldots; e_n. \tag{3}$$

The state transition diagram is shown in Figure 5.

3.2.2. Conditional Trigger Type

(i) According to the Single-Event Feature of the Event Correlation Detection. Some threat behaviors can be detected when multiple events simultaneously satisfy a certain behavioral characteristic.

The state transition relationship is

$$N = A_i \quad (i = 0, 1, 2, \ldots, n). \tag{4}$$

The state transition diagram is shown in Figure 6.

Figure 6: The state transition diagram of the single-event feature of the event correlation detection.

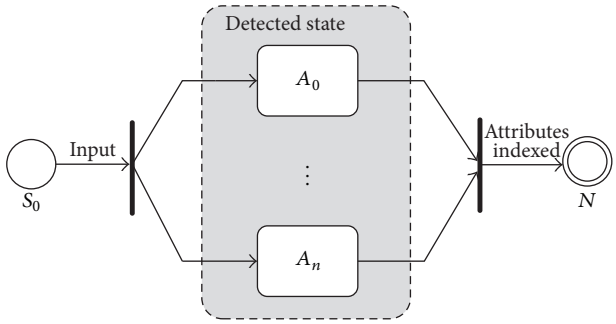


Figure 7: The state diagram of the feature of composite behavior of the correlation detection.

(ii) According to the Feature of Composite Behavior of the Correlation Detection. Some threat behaviors can be detected when multiple events simultaneously satisfy the composite behavioral characteristics.

The state transition relationship is

$$N = A_0 \mid A_1 \mid A_2 \mid \cdots \mid A_n \quad (i = 0, 1, \ldots, n). \tag{5}$$

The state transition diagram is shown in Figure 7.

3.3. Detection Algorithm Generation. According to the data packet correlation detection mode and state diagram analysis, this paper proposes the data packet correlation detection model, which can be used to detect anomaly or original data packet in intrusion detection system to improve system detection rate and reduce detection time.

3.3.1. Correlation Detection Formula

Definition 1. S_0 indicates the initial state of the detection system, S_x indicates that the system state is detected at any time, A_x represents the behavioral characteristics of the event, and N_x indicates the termination status of the detection system.

Definition 2. $N_x(S_x \mid A_x)$ indicates that the input behavior attributes A_x resulting in changes in system status, and the detection system termination status is N_x.

According to the above formula definition and state diagram, packet correlation detection formula can be as follows:

$$N_x = \begin{cases} S_0 \mid A_i & (i \in N^+) \\ (S_0 \mid S_1 \mid S_2 \mid \cdots \mid S_n) \mid A_i & (i, n \in N^+) \\ (S_0 \mid S_1 \mid S_2 \mid \cdots \mid S_n) \mid (A_0, A_1, A_2, \ldots, A_n) & (i, n \in N^+) \\ (S_0; S_1; S_2; \ldots; S_n) \mid A_i & (i, n \in N^+) \\ (S_0; S_1; S_2; \ldots; S_n) \mid (A_0, A_1, A_2, \ldots, A_n) & (i, n \in N^+). \end{cases} \tag{6}$$

3.3.2. Formula Proof

(1) $S_0 \mid A_i$, that is,

$$\forall S_{ai} = \forall S_{ax} = \forall S_{zi} \quad (a, i, x, z = 0, 1, 2, \ldots), \tag{7}$$

indicates that the detection system starts from S_0 and ends at final state N_x after a single behavior of A_i.

(2) $(S_0 \mid S_1 \mid S_2 \mid \cdots \mid S_n) \mid A_i$, that is,

$$\forall S_{ai} \neq \forall S_{zi},$$

$$\forall A_{mi} = \forall A_{mx} \tag{8}$$

$$(a, z = 1, 2, 3, \ldots), \ (i, x = 1, 2, 3, \ldots),$$

indicates that the detection system is parallel to multiple events $S_0, S_1, \ldots, S_n$ and ends at final state N_x after a single behavior of A_i.

(3) $(S_0 \mid S_1 \mid S_2 \mid \cdots \mid S_n) \mid (A_0, A_1, A_2, \ldots, A_n)$, that is,

$$\forall S_{ai} \neq \forall S_{zi},$$

$$\forall A_{mi} \neq \forall A_{mx} \tag{9}$$

$$(a, z = 1, 2, 3, \ldots), \ (i, x = 1, 2, 3, \ldots),$$

indicates that the detection system is parallel to multiple events $S_0, S_1, \ldots, S_n$ and ends at final state N_x after the composite behavior of $A_0, A_1, A_2, \ldots, A_n$.

(4) $(S_0; S_1; S_2; \ldots; S_n) \mid A_i$, that is,

$$\forall S_{ai} = \forall S_{zi},$$

$$\forall A_{mi} = \forall A_{mx} \tag{10}$$

$$(a, z = 1, 2, 3, \ldots), \ (i, x = 1, 2, 3, \ldots),$$

indicates that the detection system is serial to multiple events $S_0, S_1, \ldots, S_n$ and ends at final state N_x after a single behavior of A_i.

(5) $(S_0; S_1; S_2; \ldots; S_n) \mid (A_0, A_1, A_2, \ldots, A_n)$, that is,

$$\forall S_{ai} = \forall S_{zi},$$

$$\exists A_{mi} \neq \forall A_{mx} \tag{11}$$

$$(a, z = 1, 2, 3, \ldots), \ (i, x = 1, 2, 3, \ldots),$$

indicates that the detection system is serial to multiple events $S_0, S_1, \ldots, S_n$ and ends at final state N_x after the composite behavior of $A_0, A_1, A_2, \ldots, A_n$.

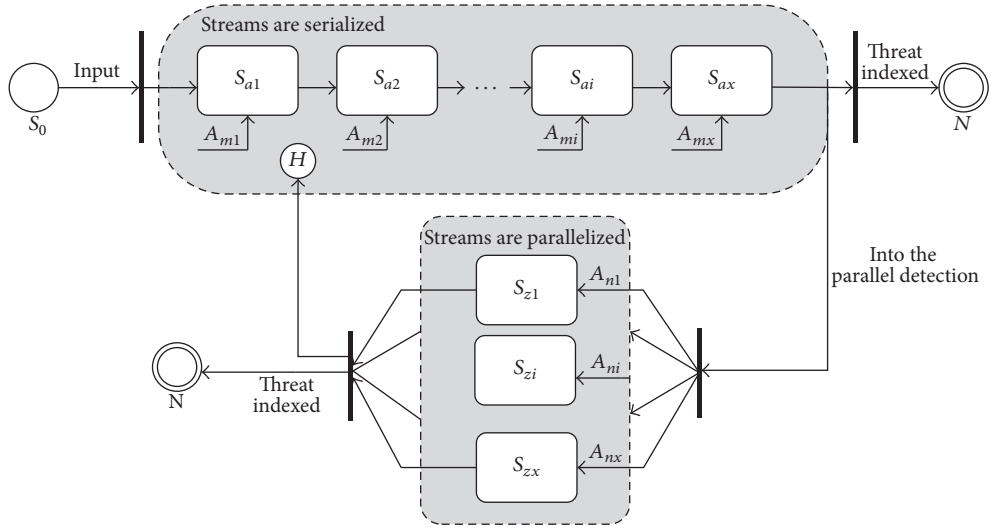


Figure 8: Correlation analysis detection model.

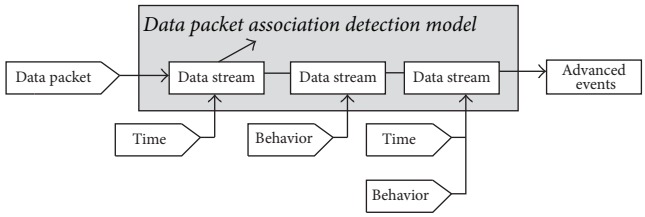


Figure 9: The detection algorithm framework.

3.3.3. Detection Algorithm Proposed. Based on the detection algorithm and the formula proposed above, this paper proposes the data packet correlation detection model. Figure 8 gives a formal representation of the data packet correlation detection model by state diagram. A key point in this figure is the use of deep analysis modules to process the associated events.

When the anomaly detection results or original data packets flow into deep analysis module in intrusion detection system, the data packet correlation detection model based on timing and behavior features can detect the events pointedly and thoroughly with the existing detection modes. Compared with traditional and single detection model, this algorithm increases detection speed and precision. Figure 9 is the detection framework of this algorithm.

In the first layer of this model, it detects timing characteristic (e.g., sniffing attacks in sequence) and behavior characteristic (e.g., as attacking continuously certain IP address). Deep analysis is in the second layer, which detects correlation detects combined with timing characteristic and behavior characteristic and aims at the detection of various persistent concealed attack behaviors. There is no need to detect data flows according to behavior feature in order and it can detect attack behaviors roundly, simplifying traditional detection modes.

4. Lightweight Intrusion Detection System Based on Correlation Analysis Detection Model

The flow diagram of deep analysis in lightweight intrusion detection system based on correlation analysis detection model is shown in Figure 10. The crucial part of the diagram is correlation detection. Firstly, it verifies ports and finds flows correlation table. Secondly, it uses correlation analysis detection model to detect and makes DPI and DFI identification. Finally, timing and behavior features are written into correlation table and these results are returned.

In this part of our paper, we will analyze and compare the traditional intrusion detection system and intrusion detection system based on correlation analysis detection model by detecting the data set that consists of 41 features in KDD1999. Then we compare the results of detection rate and detection time.

All verification work of this paper is based on KDD1999 data set. Before the experiment, we preprocess KDD1999 data set to meet experiment requirements. The environment of experiment is Window 7 operation system, and the hardware parameters are Quad-Core Intel Core i7 processor 3.2 GHz, 4096 MB RAM.

4.1. Preprocessing of Data Set. KDD1999 is a standard data set used for intrusion detection test. The KDD1999 dataset consists of a total of 5 million records, and it also provides a 10% training data set and a test data set. There are 494021 instances in training data set and 41 features in each instance, while there are 311029 instances in test data set. We divide the training data set into 5 parts: DOS attack, PROBE attack, R21 attack, U2R attack, and NORMAL. And NORMAL means normal data, excluding attack. There are 6 different kinds of attacks in DOS, 4 kinds of attacks in PROBE, 8 kinds of attacks in R21, and 4 kinds of attacks in U2R. The number of NORMAL instances is 97278 in the whole training data

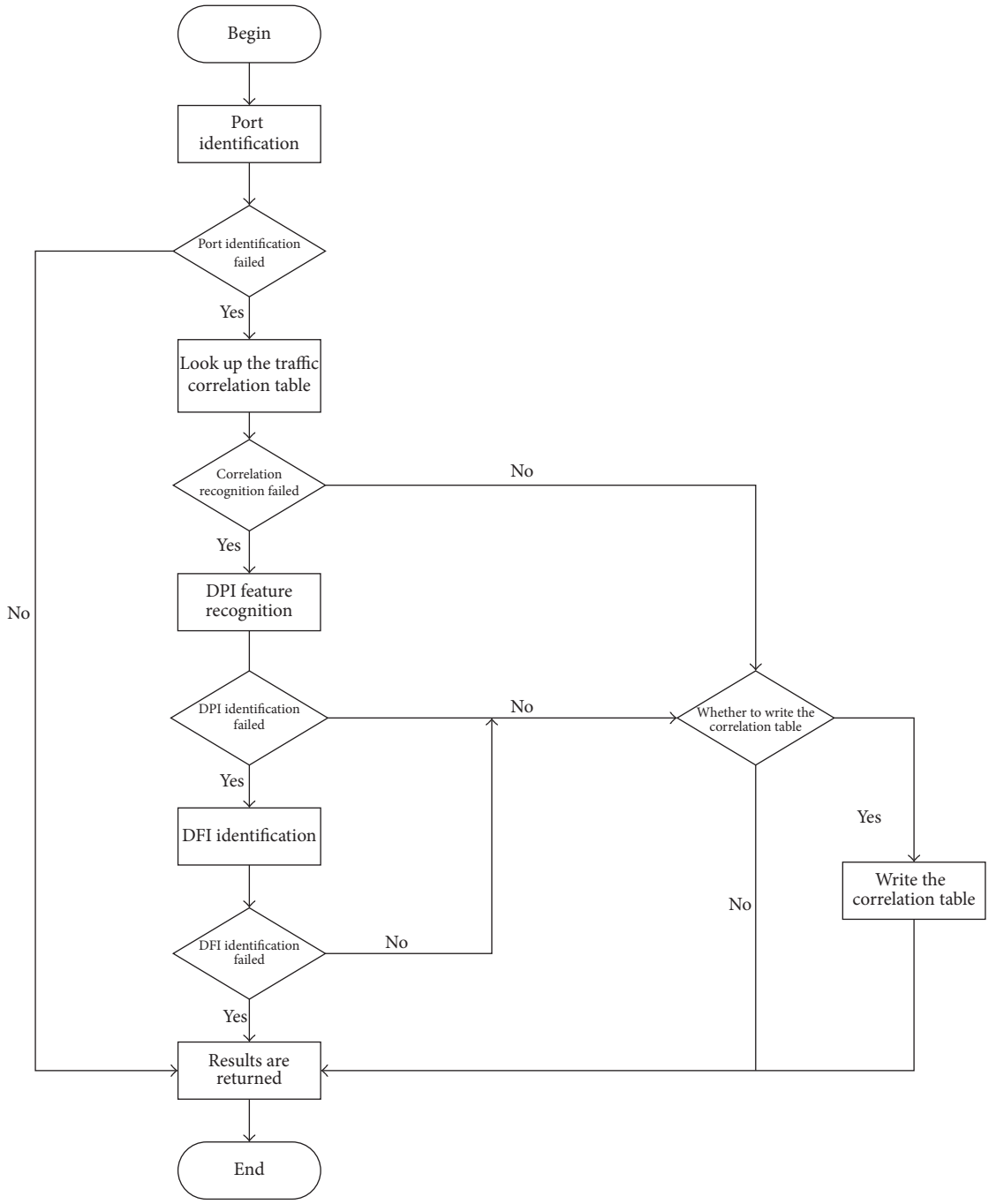


Figure 10: Intrusion detection system depth analysis process.

set; the number of DOS instances is 391458; the number of PROBE instances is 4107; the number of R21 instances is 1126, while the number of U2R instances is 52 and the sample category distribution and apart of attack types are shown in Table 1.

According to Table 1, there are 22 kinds of various attack types. As for KDD1999 test data set, we divide every type of attack into 2 parts according to the same principle: known attack and unknown attack. The known attack means the attack types that have appeared in the training data set, while the unknown attack means the attack types that have not appeared in training data set. There are 39 kinds of attack types totally in test data set: 10 types in DOS, 4 are unknown attacks; 6 types in PROBE, 2 are unknown attacks; 15 types in R21, 7 are unknown attacks; 8 types in U2R, 4 are unknown attacks. These attack types are listed in Table 1. There are 4166 instances in PROBE attack, and 2377 instances are known attack instances, accounting for 57.1% PROBE instances; 1789

Table 1: KDD1999 sample category distribution and partly attack type statistics.

Data set	Attack type	The training set	The test set
NORMAL	/	97278	60593
DOS	apache2	/	794
	back	2203	1098
	land	21	9
	mailbomb	/	5000
	neptune	107201	58001
	pod	264	87
	processtable	/	759
PROBE	ipsweep	1247	306
	mscan	/	1053
	nmap	231	84
	portsweep	1040	354
	saint	/	736
	satan	1589	1633
R2L	ftp_write	8	3
	imap	12	1
	named	/	17
	phf	4	2
	sendmail	/	17
U2R	httptunnel	/	158
	loadmodule	9	2
	perl	3	2
	ps	/	16
	rootkit	10	13
	sqlattack	/	2
Total	39	494021	311029

Table 2: The experiment result.

	Detection rate (%)	Detection time (s)	Known attack detection rate (%)	Unknown attack detection rate (%)
Traditional IDS	79	182	87	71
IDS based on the correlation model	92	156	95	89

instances are known attack instances, representing 42.9%. The figures for instances of known and unknown attacks are in Table 1.

4.2. Experimental Program. After the analysis and process of KDD 1999 data set in the section above, the number of instances in training set is much larger. We know that, in order to make experiment operation more convenient, we should select data to reduce the number of instances. We sample the DOS, PROBE, R21, U2R, and NORMAL in training set randomly and respectively and ensure the consistency of these samples and the original sample. Then we combine these 5 new samples and form a new training data set. These 5 samples are the combination of NORMAL and DOS, the combination of NORMAL and PROBE, the combination of NORMAL and R21, the combination of NORMAL and U2R, and the combination of NORMAL, DOS, PROBE, R21, and U2R. The instance of 5 new training sets is 98630. These 5 training sets are flowed into the traditional intrusion detection system and intrusion detection system based on data packet correlation detection model. We compare the performances based on the same data resource by comparing detection rate, detection time, and so on.

4.3. Experimental Results and Analysis. The experiment result is shown in Table 2; it is obvious to get that the detection rises sharply in intrusion detection system based on the data packet correlation detection model and the detection time decreases, promoting the efficiency of detection system.

Therefore, for intrusion detection system based on the correlation analysis detection model in this paper, the detection rate of known and unknown attacks is high, improving the performance of intrusion detection system.

5. Conclusions

In this paper, we build a high-performance correlation analysis detection model, which aims to resolve the low detection rate and slow detection speed.

For the intrusion detection system, we put forward a kind of universal network intrusion detection framework. Meanwhile, we analyze and compare the current common correlation intrusion detection modes. Finally, we propose a data packet correlation detection model and algorithm based on the data packet timing and behavior characteristics. In the experiments, this kind of correlation detection model has improvement in performance than former. In this paper the present popular intrusion detection system has good practical value.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China Project (61302087).

References

[1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[2] H. Liao J, C. Lin H R, Y. Lin C et al., "Intrusion detection system: a comprehensive review," *Journal of Network Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[3] S. Benferhat, A. Boudjelida, K. Tabia, and H. Drias, "An intrusion detection and alert correlation approach based on revising probabilistic classifiers using expert knowledge," *Applied Intelligence*, vol. 38, no. 4, pp. 520–540, 2013.

[4] J. Liang, "Research on network intrusion detection system based on snort," *Information Security and Technology*, vol. 6, no. 2, pp. 37-38, 2015.

[5] W. Bul'ajoul, A. James, and M. Pannu, "Improving network intrusion detection system performance through quality of service configuration and parallel technology," *Journal of Computer and System Sciences*, vol. 81, no. 6, article 2856, pp. 981–999, 2015.

[6] A. Das, D. Nguyen, J. Zambreno, G. Memik, and A. Choudhary, "An FPGA-based network intrusion detection architecture," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 118–132, 2008.

[7] D. Abdelouahid and B. Abdelghani, "Multivariate correlation analysis and geometric linear similarity for real-time intrusion detection systems," *Security & Communication Networks*, vol. 8, no. 7, pp. 1193–1212, 2015.

[8] M. Choraś, Ł. Saganowski, R. Renk, and W. Hołubowicz, "Statistical and signal-based network traffic recognition for anomaly detection," *Expert Systems the Journal of Knowledge Engineering*, vol. 29, no. 3, pp. 232–245, 2012.

[9] S. Shin, T. Kwon, G.-Y. Jo, Y. Park, and H. Rhy, "An experimental study of hierarchical intrusion detection for wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 744–757, 2010.

[10] W. Wang, J. Liu, G. Pitsilis et al., "Abstracting massive data for lightweight intrusion detection in computer networks," *Information Sciences*, 2016.

[11] B. Luo and J. Xia, "A novel intrusion detection system based on feature generation with visualization strategy," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4139–4147, 2014.

[12] Y. Yang, H. Huang, S. Shen et al., "Intrusion detection based on incremental ghsom neural network model," *Computer Journal*, no. 5, pp. 1216–1224, 2014.

[13] R. Sommer and V. Paxson, "Outside the closed world: on using machine learning for network intrusion detection," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 305–316, IEEE Computer Society, 2010.

[14] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden naïve bayes multiclass classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13492–13500, 2012.

[15] U. Fiore, F. Palmieri, A. Castiglione, and A. de Santis, "Network anomaly detection with the restricted boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[16] S.-W. Lin, K.-C. Ying, C.-Y. Lee, and Z.-J. Lee, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection," *Applied Soft Computing Journal*, vol. 12, no. 10, pp. 3285–3290, 2012.

[17] I. Friedberg, F. Skopik, G. Settanni et al., "Combating advanced persistent threats: from network event correlation to incident detection," *Computers & Security*, vol. 48, no. 7, pp. 35–57, 2015.

[18] F. Amiri, M. M. R. Yousefi, C. Lucas et al., "Mutual information-based feature selection for intrusion detection systems," *Journal of Network & Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[19] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers and Security*, vol. 29, no. 1, pp. 124–140, 2010.

[20] M. A. Ambusaidi, X. He, P. Nanda et al., "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[21] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

[22] Z. Tan, A. Jamdagni, X. He et al., "Denial-of-service attack detection based on multivariate correlation analysis," *Neural Information Processing*, vol. 7064, no. 2, pp. 756–765, 2013.

[23] W. Wang, T. Guyet, R. Quiniou, M.-O. Cordier, F. Masseglia, and X. Zhang, "Autonomic intrusion detection: adaptively detecting anomalies over unlabeled audit data streams in computer networks," *Knowledge-Based Systems*, vol. 70, pp. 103–117, 2014.

[24] W. Wang, X. Guan, and X. Zhang, "Processing of massive audit data streams for real-time anomaly intrusion detection," *Computer Communications*, vol. 31, no. 1, pp. 58–72, 2008.

[25] R. Shittu, A. Healing, R. Ghanea-Hercock, R. Bloomfield, and M. Rajarajan, "Intrusion alert prioritisation and attack detection using post-correlation analysis," *Computers and Security*, vol. 50, pp. 1–15, 2015.

[26] S. Salah, G. Ndez, J. Az-Verdejo et al., "Survey A model-based survey of alert correlation techniques," *Computer Networks the International Journal of Computer & Telecommunications Networking*, vol. 57, no. 5, pp. 1289–1317, 2013.

[27] A. A. Amaral, B. B. Zarpelão, L. D. S. Mendes et al., "Inference of network anomaly propagation using spatio-temporal correlation," *Journal of Network & Computer Applications*, vol. 35, no. 6, pp. 1781–1792, 2012.

[28] P. Xiao, W. Y. Qu, H. Qi, and Z. Y. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, pp. 66–74, 2015.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8596893, 9 pages
https://doi.org/10.1155/2017/8596893

Research Article

Multiview Community Discovery Algorithm via Nonnegative Factorization Matrix in Heterogeneous Networks

Wang Tao and Liu Yang

PLA Information Engineering University College of Information Systems Engineering, Zhengzhou, China

Correspondence should be addressed to Wang Tao; yjswangtao@163.com

Received 16 October 2016; Accepted 19 February 2017; Published 7 May 2017

Academic Editor: Liu Yuhong

With the rapid development of the Internet and communication technologies, a large number of multimode or multidimensional networks widely emerge in real-world applications. Traditional community detection methods usually focus on homogeneous networks and simply treat different modes of nodes and connections in the same way, thus ignoring the inherent complexity and diversity of heterogeneous networks. It is challenging to effectively integrate the multiple modes of network information to discover the hidden community structure underlying heterogeneous interactions. In our work, a joint nonnegative matrix factorization (Joint-NMF) algorithm is proposed to discover the complex structure in heterogeneous networks. Our method transforms the heterogeneous dataset into a series of bipartite graphs correlated. Taking inspiration from the multiview method, we extend the semisupervised learning from single graph to several bipartite graphs with multiple views. In this way, it provides mutual information between different bipartite graphs to realize the collaborative learning of different classifiers, thus comprehensively considers the internal structure of all bipartite graphs, and makes all the classifiers tend to reach a consensus on the clustering results of the target-mode nodes. The experimental results show that Joint-NMF algorithm is efficient and well-behaved in real-world heterogeneous networks and can better explore the community structure of multimode nodes in heterogeneous networks.

1. Introduction

Community structure is an important feature of real-world networks as it is crucial for us to study and understand the functional characteristics of the real complex systems. With the fast growth of Internet and computational technologies in the past decade, many data mining applications have advanced swiftly from the simple clustering of one data type to the multiple types, which usually involved high heterogeneity, such as the interrelations of users, videos, pictures, and web page in web networks (shown in Figure 1). Those networks with multiple modes/dimensions are called heterogeneous network in this work. Unlike homogeneous networks that only contain one kind of nodes and have explicit community structure, the community structures of heterogeneous networks are usually obscure and complicated, which are owing to the coexistence of multimode or multidimensional interactions. Therefore, it is challenging to effectively integrate the information of multiple dimensions/modes to discover the hidden community structure underlying heterogeneous interactions.

There are a number of problems for traditional clustering methods to mine the community structure in heterogeneous networks. First, heterogeneous networks contain different types of nodes and relationships. Processing and interpreting them in a unified way present a major challenge. Second, various data types are related to each other. Tackling each type independently will lose the mutual information between those interactions, which are essential to gain a full understanding of heterogeneous networks. Consequently, matrix-factorization-based clustering has emerged as an effective approach for clustering problems in high-dimensional datasets. In [1], it is shown that nonnegative matrix factorization outperforms spectral methods in document clustering, achieving higher accuracy and efficiency.

In this paper, we adopt multiview learning as a tool to reveal the communities in heterogeneous networks, because it has powerful interpretability and applicability for data

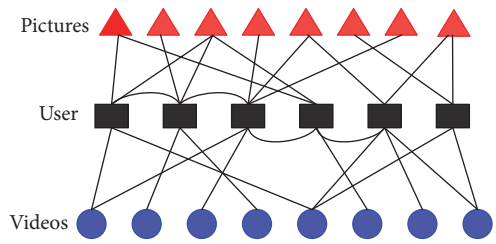


Figure 1: An illustrative example via web networks.

clustering. So we present a joint nonnegative matrix factorization (Joint-NMF) solution to detect community in heterogeneous networks. To summarize, the main contributions of this work includes the following: (1) we construct the bipartite graph model of heterogeneous networks, which efficiently incorporates the multimode or multidimensional information, to enhance the community detection in heterogeneous networks. (2) We propose an optimal algorithm for the iterative procedure of joint matrix factorization. Computationally, Joint-NMF clustering is more efficient and flexible than graph-based models and can provide more intuitive clustering results. In particular, it provides mutual information between each network graph to realize the collaborative learning of different classifiers and makes all the classifiers tend to reach a consensus on the clustering results of heterogeneous networks.

The remainder of the paper is organized as follows. In Section 2, we demonstrate the related work about corresponding domain and define the bipartite graph model of heterogeneous networks. In Section 3, we formulate the multiview method via Joint-NMF for community detection in heterogeneous networks and present an optimal algorithm to achieve fast convergence. Then we test our algorithm on a variety of real heterogeneous networks and present the experimental results in Section 4. Finally, Section 5 concludes the paper.

2. Preliminary

2.1. Related Work. In the past years, the research of community discovery in heterogeneous networks has attracted more and more attention of researchers. Among these methods, matrix factorization effectively reflects the community structure of the networks and promises a meaningful community interpretation that is independent of the network topology. In addition to a quantification of how strongly each node participates in its community, nonnegative matrix factorization (NMF) does not suffer from the drawbacks of modularity optimization methods [2], such as the resolution limit [3]. Nguyen et al. [4] used nonnegative matrix factorization with I-divergence as the cost function and introduce two approaches which are, respectively, applied to the directed and undirected networks. Based on the importance of each node when forming links in each community, He et al. [5] use nonnegative matrix factorization to form a generative model, taking it as an optimization problem to discover the structure of link communities. Chen et al. [6] presented a semisupervised community discovery algorithm based on NMF. It introduced the a priori knowledge as constraints into the heterogeneous networks and reconstructed the feature matrix to detect communities. Tang et al. [7] introduced the concept of modularity optimization into the heterogeneous network and integrated the network snapshots in a time period and then obtained the community partition with maximum at that moment. Wang et al. [8] took the internal connections as the graph regularization constraint and utilize tri-NMF model to improve the performance of community detection in bipartite networks.

Due to multidimensional nodes and special link patterns in heterogeneous networks, it is more suitable to mine the special community structure via semisupervised methods. However, most community detection methods still focus on homogeneous networks and might not work well in heterogeneous networks. There are two main reasons: first, heterogeneous data contain different types of relations. Processing and interpreting them in a unified way present a major challenge. Second, the special link patterns of heterogeneous networks greatly limit the effectiveness of these methods, which tend to cluster the multimode nodes by constructing the node or edge similarities of them, as they did in the homogeneous networks. But, for heterogeneous networks, the similarities among one-mode nodes sometimes can only be defined by the nodes of the other mode. That made these methods unable to keep working well in heterogeneous networks. In summary, most works are derived based on the graph model, which requires solving eigenvalue-problem. Computationally, they are inefficient and inapplicable to large-scale datasets. Moreover, they are completely unsupervised and ignore the inherent complexity and diversity of heterogeneous networks.

Recent works [9–11] have shown that multiview learning in multimode datasets can effectively improve the clustering performance in the sense that clustering can make full use of the dual interdependence between multiple nodes to discover certain hidden community structures. In this work, we present a semisupervised method (Joint-NMF) to incorporate multimode/multidimension information for unity discovery. In the proposed methodology, users are able to provide constraints on the target mode, specifying the multiple connecting relationships in each bipartite graph. Our goal is to improve the quality of community structure by multiview learning in all modes of nodes and linking. Using an optimal iterative procedure, we then perform joint-factorizations of the bipartite graph matrices to obtain the consensus of network partition and finally infer the target-mode clusters while simultaneously deriving the communities of related feature nodes. In addition, due to the fact that NMF-based methods often require the community numbers of networks to be specified beforehand, several methods [10, 12, 13] have been developed to solve this problem. Due to the simplicity and practicability of the existing method in [4], here we choose it to get the community numbers.

2.2. Model Formulation. Both the multimode and multidimensional networks can be modeled as bipartite graphs, which completely describe the diverse properties and

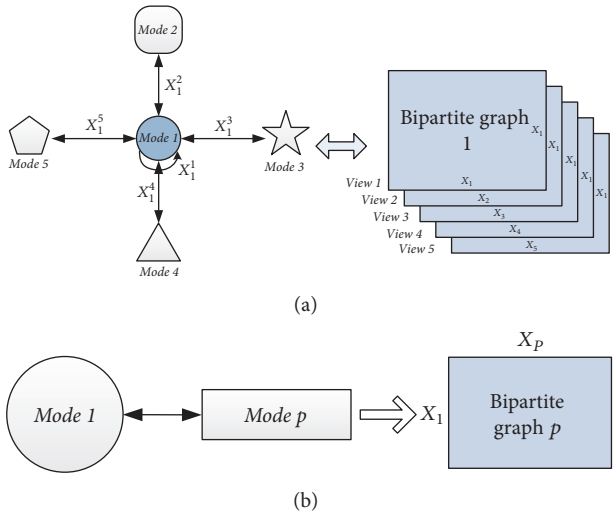


Figure 2: (a) Transforming web networks into the bipartite graph model. (b) According to the bipartite graph model, the relationships between modes 1 and p are expressed as graph X_1^p.

characteristics of the multimode connections. In this way, the heterogeneous network can be regarded as a comprehensive depiction of multimode nodes, described by a series of subbipartite graphs. Each bipartite graph represents the relationship between a special kind of node and the target nodes and contains the structural feature of the heterogeneous network in its own perspective. Compared to processing each subgraph independently, combining multiple bipartite graphs would undoubtedly be more effective and accurate for community discovery. For example, the document can be clustered through both semantics and reference relationship, and multimedia resources can be through the content annotation and the user preferences.

By means of the multiview learning, we treat each bipartite graph as one independent feature set of the target nodes. In this way, heterogeneous networks can be depicted in multifaceted, different perspectives simultaneously. As shown in Figure 1, the web network can be expressed with the triple vector. Assuming users as the target mode, pages and tags are the two-dimensional feature space that reflects the community structure of user nodes. Specifically, the bipartite graph $X_{\text{User}}^{\text{Picture}}$ shows the users' feature in the picture dimension, and $X_{\text{User}}^{\text{Video}}$ indicates the users' feature in the video dimension. Based on the bipartite graph model (shown in Figure 2), heterogeneous network can be decomposed into a series of bipartite graphs $\{X_1^1, X_1^2, \ldots, X_1^p\}$, where X_1^p indicates the relationships between mode 1 and mode p.

In this work, we take the core mode nodes as target nodes and give priority to the community division of target nodes. Due to the core position of target nodes in heterogeneous networks, community distribution of the other nodes is exclusively conducted and decided by the community structure of target nodes. Through grouping the target nodes in different communities, our method simultaneously divides the connecting nodes of the other mode into the corresponding communities. Through the bipartite graphs model, each mode's nodes can naturally be split into different graphs, any of which suffices for mining knowledge. Observing that these bipartite graphs often provide compatible and complementary information, it becomes natural for one to integrate them together to obtain better performance rather than relying on a single view. Based on the bipartite graph model, we can use multiview learning for seamlessly integrating multiple node information to discover the underlying community structure in heterogeneous networks.

3. Multiview Algorithm via Joint-NMF for Community Detection in Heterogeneous Networks

In this section, based on multiview learning, we propose a joint nonnegative factorization matrix algorithm for community detection. This method adopts semisupervised learning to integrate multiple bipartite graphs in heterogeneous networks and extends semisupervised learning from single graph to multiview graphs. Based on the collaborative learning between different modes (graphs), the multiview learners finally obtain the consensus on the clustering results of the target nodes, thus jointly promoting the performance of community discovery. For convenience, we present in the Notations the important notations used in this paper.

3.1. Objective Function of Multiview Learning via Joint-NMF. For the original NMF framework, it just considers the intertype information of 1-mode nodes. Such formulation assumes each subnetwork to be independent and fails to model the heterogeneous networks in a unified way. Recently, some researchers [9, 10] have found that multiview learning on multiple bipartite graphs is well applied to heterogeneous networks for community clustering, because it can promote the performance of the intrinsic structure discovery in multimode networks. As a result, by constructing the bipartite graphs of heterogeneous networks, the optional intertype information of different modes of nodes is incorporated into Joint-NMF. More importantly, we can exploit the mutual information from multidimensional spaces to group like-minded nodes from different graph perspectives, thus strengthening the community detection in heterogeneous networks.

For multiview learning, the bipartite graphs of different mode have conditional independence. It means that all the independent learners under different graphs can not make the wrong decision in the same time. Therefore, our semisupervised learning method can effectively integrate bipartite graph information and unlabeled information and adopt cooperative learning between multiview graphs to surmount the obstacle of complexity and diversity in heterogeneous datasets. In this way, our method realizes the information complementation of different information graph, thereby enhancing the overall performance of community discovery in heterogeneous networks.

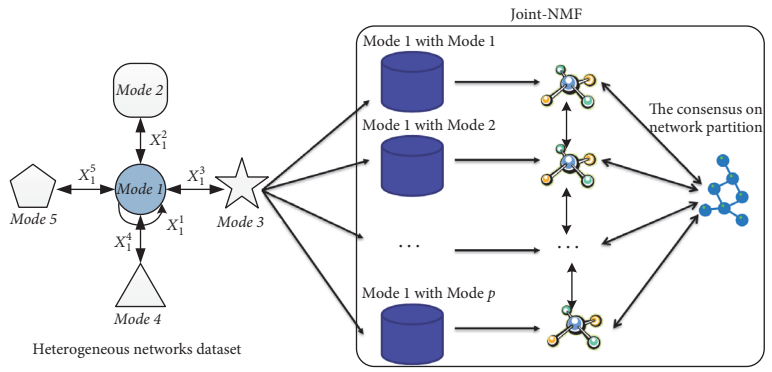


Figure 3: The Joint-NMF model for community detection in heterogeneous networks with multiview learning.

In order to group the relevant target-mode nodes and the corresponding nodes of other mode into the same community, the following objective function is used to measure the accuracy and smoothness of clustering results:

$$\min \quad \sum_{i=1}^{M} \left\| X_1^i - W^i \left(H^i\right)^T \right\|_F^2 + \sum_{j=1}^{M} \lambda_j \left\| H^j - H^* \right\|_F^2 \qquad (1)$$
$$\text{s.t.} \quad W^i, H^i, H^* \geq 0,$$

where $W^i \in \mathbb{R}^{m\times r}$ and $H^i \in \mathbb{R}^{n\times r}$, respectively, denote the basis matrix and coefficient matrix decomposed from graph X_1^i. m is associated with the node number of mode 1, and n is the node number of model i, and let r be the preset community number. In particular, it is worthwhile to note that mode 1 is regarded as the target node in this paper. The internal connection matrix X_1^1 has also been incorporated for nonnegative matrix factorization.

For each single bipartite graph $X_1^i \in \mathbb{R}^{m\times n}$ (Figure 3), according to the principle of NMF, we can minimize the objective function $\min \|X_1^i - W^i(H^i)^T\|_F^2$ to obtain the ith dimensional clustering results of the target nodes. And, for heterogeneous network, it is indispensable to minimize the sum of fitting errors for revealing the community structure of target nodes in the multidimensional/multimode space.

However, Joint-NMF is also subject to several problems such as slow convergence and large computation. Moreover, heterogeneous networks have more complicated connecting relationships between multiple modes/multiple dimensions, which further limits the application and effectiveness of NMF to explore the hidden community structures. Aiming at these problems, our work mainly optimizes the iterative solutions of NMF from the two aspects:

(1) To simplify the iterative procedure, a matrix $Q_{r\times r}$ is introduced that satisfies the condition

$$WH^T = \left(WQ^{-1}\right)\left(QH^T\right). \qquad (2)$$

(2) Inspired by [14, 15], we incorporate the idea of multiview learning into joint matrix factorization, which can effectively extend semisupervised learning from single graph to multiple graphs, from single mode to multiple-mode nodes. Moreover, our method adds some essential constraints on the matrix factorization, which can ensure the uniqueness and accuracy of the results in network partition.

3.2. Optimization Algorithm of Joint-NMF. Assuming mode 1 as the target mode in heterogeneous network, the other modes are all connected with the target mode. To achieve an accuracy network partition with joint nonnegative matrix factorization, it must satisfy the condition

$$D\left(H^i, H^*\right) = \left\| H^i - H^* \right\|_F^2, \quad 0 < i \leq p, \qquad (3)$$

where H^* denotes the coefficient matrix of the target nodes, which is concluded under the multiview learning of all the modes $\{X_1^1, X_1^2, \ldots, X_1^p\}$. H^* indicates the final consensus of community partition on the target node and the other nodes.

First, we construct the objective function of Joint-NMF, which mainly comprises two terms: the first term is the standard NMF approximation of the objective function, and the second one is a penalty term about the deviation from the consensus H^*. In particular, by means of semisupervised learning in multiple bipartite graphs, it is realized that the collaborative learning between multiple modes or multiple dimensions is applied to community structure mining in heterogeneous networks. In this way, the target nodes can be clustered as consistent as possible, and then we automatically obtain the cluster results of the other nodes with maximum consistency.

Applying the regularizations (see (3)) in (1), the objective of our Joint-NMF approach is transformed to minimize

$$\min \quad \sum_{i=1}^{M} \left\| X_1^i - W^i \left(H^i\right)^T \right\|_F^2 + \sum_{j=1}^{M} \lambda_j \left\| H^j - H^* \right\|_F^2 \qquad (4)$$
$$\text{s.t.} \quad W^i, H^i, H^* \geq 0,$$

where mode 1 is the target nodes and the objective combines the internal connection of mode 1 X_1^1 and all other connecting graphs $\{X_2^1, \ldots, X_p^1\}$ related to mode 1. λ_j is mainly used to adjust the weight of bipartite graph X_j^1, and it also reflects the

importance of the corresponding linking mode in networks, $0 < \lambda_j < 1$. When $\lambda_j = 0$, (4) is transformed into an unsupervised NMF function.

In order to optimize the procedure of matrix factorization, we construct a special diagonal matrix

$$Q^i = \mathrm{Diag}\left(\sum_{j=1}^{M} W_{j,1}^i, \sum_{i=1}^{M} W_{j,2}^i, \ldots, \sum_{i=1}^{M} W_{j,r}^i\right), \tag{5}$$

where $\mathrm{Diag}(\cdot)$ denotes the diagonal matrix operations. According to (4), the objective function can be transformed into the following minimization problem:

$$\begin{aligned} & O \\ & = \sum_{i=1}^{M} \left\| X_1^i - W^i \left(H^i\right)^T \right\|_F^2 + \sum_{j=1}^{M} \lambda_j \left\| H^j Q^j - H^* \right\|_F^2 \\ \text{s.t.} \quad & \forall 1 \le i \le p+1, \\ & W^i, H^i, H^* \ge 0. \end{aligned} \tag{6}$$

Since Joint-NMF (see (4)) is a nonconvex function about factor matrices W, H, it is difficult to obtain the global optimal solution directly. As a result, we propose an alternative iterative update solution, which iterates with the following two steps sequentially to reach the fast convergence.

Step 1. Fixing H^*, calculate W^i and H^i to minimize the objective function.

Step 2. Fixing W^i and H^i, calculate H^* to minimize the objective function.

In this way, the alternative iterative update solution fixes one relevant variable with the latest value, and thus the minimizing objective (see (6)) is transformed into a convex optimization problem about some single variable. By means of alternating iterative update, we would get the local extremum solution or stable solution finally. Thus, our multiplicative update procedure can effectively be applied to multiple matrix factorization and speed up the convergence process.

To minimize the objective function, it is necessary to decompose each bipartite graph separately to converge. For one single graph X_1^i, its specific objective function is regarded as

$$O^i = \left\| X_1^i - W^i \left(H^i\right)^T \right\|_F^2 + \lambda_i \left\| H^i Q^i - H^* \right\|_F^2, \quad W, H \ge 0; \tag{7}$$

when H^* is fixed, for any given graph X_1^i, the results of W^i and H^i do not rely on the calculation of other graphs. Assuming Ψ as the Lagrange multipliers that constrain $W \ge 0$, the Lagrangian function of (6) can be simplified as $L = O + \mathrm{Tr}(\Psi W^i)$, and $\mathrm{Tr}(\cdot)$ denotes the matrix trace. Accordingly, L can be rewritten as

$$L_1 = \mathrm{Tr}\left(WH^T HW^T - 2XHW^T\right) + \lambda_i R + \mathrm{Tr}\left(\Psi W\right), \tag{8}$$

where $R = \mathrm{Tr}(HQQ^T H^T - 2HQ(H^*)^T)$ includes the regularization term $\|HQ - H^*\|_F^2$. Introducing the diagonal matrix Q constructed in (5), R can be rewritten as

$$\begin{aligned} R = & \sum_{j=1}^{r} \sum_{k=1}^{n} \left(H_{j,k} \sum_{i=1}^{M} W_{i,k} \sum_{i=1}^{M} W_{i,k} H_{j,k} \right) \\ & - \sum_{j=1}^{r} \sum_{k=1}^{n} \left(H_{j,k} \sum_{i=1}^{M} W_{i,k} H_{j,k}^* \right). \end{aligned} \tag{9}$$

By setting the derivative of R with respect to W, we obtain

$$\frac{\partial R}{\partial W_{i,k}} = 2\left(\sum_{i=1}^{M} W_{i,k} \sum_{j=1}^{M} H_{j,k}^2 - \sum_{l=1}^{M} H_{l,k} H_{j,k}^* \right). \tag{10}$$

According to Karush-Kuhn-Tucker (KKT) condition in [13], set the derivative of L_1 with respect to U:

$$\begin{aligned} & \frac{\partial L_1}{\partial U} = -2XV + 2UV^T V + \lambda_i P + \Psi = 0 \\ & \Psi_{i,k} U_{i,k} = 0, \quad \forall 1 \le i \le M, \ 1 \le k \le r. \end{aligned} \tag{11}$$

Based on KKT optimization condition, the solution $W_{i,k}$ is obtained by

$$W_{i,k} \longleftarrow W_{i,k} \frac{(XH)_{i,k} + \lambda_i \sum_{j=1}^{M} H_{j,k} H_{j,k}^*}{(WH^T H)_{i,k} + \lambda_i \sum_{l=1}^{M} W_{l,k} \sum_{j=1}^{M} H_{j,k}^2}. \tag{12}$$

If the initialization $W_{i,k} > 0$, it can be concluded that $W_{i,k}$ will remain nonnegative in the subsequent iteration.

Fixing H^* and W^i, use the diagonal matrix Q to normalize the column vector of matrix W, and then we obtain

$$\begin{aligned} W & \longleftarrow WQ^{-1}, \\ H & \longleftarrow HQ, \end{aligned} \tag{13}$$

where the normalization process does not change the numerical value of W, H. Assuming Φ as the Lagrange multipliers that constrain $H \ge 0$, the Lagrangian function of (8) can be rewritten as

$$\begin{aligned} L_2 = & \mathrm{Tr}\left(WH^T HW^T - 2XHW^T\right) \\ & + \lambda_i \mathrm{Tr}\left(HH^T - 2H\left(H^*\right)^T\right) + \mathrm{Tr}\left(\Phi H\right). \end{aligned} \tag{14}$$

Similarly, according to Karush-Kuhn-Tucker (KKT) condition in [13], set the derivative of L_2 with respect to H:

$$\begin{aligned} & \frac{\partial L_2}{\partial H} = 2HW^T W - 2X^T W + 2\lambda_i \left(H - H^*\right) + \Phi \\ & \qquad = 0, \\ & \Phi_{j,k} H_{j,k} = 0, \quad \forall 1 \le j \le p+1, \ 1 \le k \le r. \end{aligned} \tag{15}$$

Hence, the iterative update solution of $H_{j,k}$ is regarded as

$$H_{j,k} \longleftarrow H_{j,k} \frac{\left(X^T W\right)_{j,k} + \lambda_i H_{j,k}^*}{\left(HW^T W\right)_{j,k} + \lambda_i H_{j,k}}. \tag{16}$$

Input: Network $\{X_1^1, X_1^2, \ldots, X_1^p\}$, $\{\lambda_1, \lambda_2, \ldots, \lambda_p\}$, r;
Output: Consensus coefficient matrix H^*
(1) Normalize $X_1^i \in \{X_1^1, X_1^2, \ldots, X_1^p\}$; //normalize all the dataset matrices
(2) Initialize W^i, H^i, H^*; //$1 \le i \le p$
(3) % the iterative procedure of Joint-NMF%
(4) **repeat**
(5) **for** each $i \in N$ **do**
(6) **repeat**
(7) Fixing H^* and H^i, update W^i by Eq. (12);
(8) Normalize W^i and H^i by Eq. (13);
(9) Fixing $H^*$and W^i, update H^i by Eq. (18);
(10) **until** Eq. (7) converges.
(11) **end for**
(12) Fixing W^i and H^i, update H^* by Eq. (18);
(13) **until** Eq. (6) converges.
(14) **return** W, H, H^*.

Algorithm 1: Joint-NMF.

After getting the matrices W^i and H^i, according to KTT condition, set the derivative of O with respect to H^*:

$$\frac{\partial O}{\partial H^*} = \frac{\partial \sum_{i=1}^{p} \lambda_i \left\| H^i Q^i - H^* \right\|_F^2}{\partial H^*} = -2\lambda_i \sum_{i=1}^{p} \left(H^i - H^* \right) = 0. \tag{17}$$

To minimize the objective O, the iterative update solution of $H_{j,k}$ is regarded as

$$H^* = \frac{\sum_{i=1}^{p} \lambda_i H^i Q^i}{\sum_{i=1}^{p} \lambda_i} \ge 0. \tag{18}$$

Repeat the above iteration of matrix factorization, and update the factor matrixes W, H, and H^* continuously, until the objective function tends to converge or reaches the maximum iteration number.

After iterations, we can infer the community membership of multiple nodes based on the Joint-NMF results. For simplicity, the community indices are determined by taking the maximum of each column in H^* (the target nodes) and H^i (the other mode nodes). Note that once we obtain the consensus matrix H^*, the cluster label of mode i could be computed from H^i. The detailed procedure is illustrated in Algorithm 1.

3.3. Algorithm Convergence and Complexity. Here we prove the theoretical convergence of Joint-NMF algorithm.

Proposition 1. *Given a bipartite graph $X_1^i \in \mathbb{R}^{m \times n}$ and its initialization factor matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{n \times k} \ge 0$, the objective function (9) decreases monotonically under the alternative iterative update rules (see (12) and (18)).*

Proof. The proof of the *proposition is similar to* the convergence proof of nonnegative matrix triple-factorization in [16]. Moreover, recent studies [17] found the following: despite the alternating iterative update may fail to converge to a stable point, but the improved iterative rules (see (12) and (18)) can guarantee Joint-NMF algorithm can converge to a local extremum point. □

In our algorithm, W and H are sparse matrices, and the computation of them only involves vector norm enumeration without matrix multiplication, and thus it is more computationally efficient. Moreover, instead of minimizing each matrix factor optimally with time-consuming multiplications of large matrices, Joint-NMF transforms the original heterogeneous networks into some bipartite graphs requiring much fewer matrix multiplications and effectively optimizes the iterative procedure with faster convergence and lower computational complexity.

The running time of our algorithm is mainly consumed in the alternative iterative procedure. For single network graph X_1^i, the complexity of matrix factorization is $O(trmn)$, where t is the iterative number of algorithms and r is the preset community number. As a result, the computational complexity for Joint-NMF method is $O(ptrmn)$.

4. Experimental Results

In this section, the experiments use a series of real networks to validate the algorithms' performance. Real networks are always more irregular and various than synthetic networks and have more complex community structures. Here we choose 4 popular real heterogeneous networks in different sizes: WebKB [18], Newsgroups [19], Cora [20], and Last.fm [21]. For all the networks, we compare the experimental results with other 4 well-known algorithms of community detection: *K*means [12], NMF [13], SS-NMF [6], and PMM [22]. All the experiments are performed on an Intel Core2 Duo 2.0 GHz PC with 2 GB RAM, running on Windows 7.

Table 1: The average execution time of the 5 community detection methods on the real networks.

Algorithm	Execution time (seconds)			
	WebKB	Newsgroups	Cora	Last.fm
*K*means	6.57×10^3	2.06×10^4	2.15×10^3	8.51×10^5
NMF	1.07×10^4	4.41×10^4	5.36×10^3	1.48×10^6
SS-NMF	1.81×10^4	6.33×10^4	8.61×10^4	2.18×10^6
PMM	2.15×10^4	7.53×10^4	6.13×10^4	2.96×10^6
Joint-NMF	1.51×10^4	4.52×10^4	5.01×10^4	1.68×10^6

Table 2: Clustering accuracy ± standard deviation of the 5 community detection methods on the real networks.

Algorithm	Clustering accuracy (%)			
	WebKB	Newsgroups	Cora	Last.fm
*K*means	59.4 ± 0.05	64.8 ± 0.07	61.2 ± 0.01	47.6 ± 0.05
NMF	67.6 ± 0.09	81.1 ± 0.03	63.9 ± 0.02	54.7 ± 0.02
SS-NMF	73.8 ± 0.07	91.3 ± 0.04	72.7 ± 0.06	58.6 ± 0.04
PMM	71.5 ± 0.06	84.1 ± 0.08	67.8 ± 0.03	56.0 ± 0.08
Joint-NMF	78.1 ± 0.02	94.9 ± 0 .01	75.1 ± 0.09	61.5 ± 0.01

Table 3: NMI ± standard deviation of the 5 community detection methods on the real networks.

Algorithm	NMI (%)			
	WebKB	Newsgroups	Cora	Last.fm
*K*means	37.5 ± 0.02	61.2 ± 0.04	49.1 ± 0.08	47.6 ± 0.02
NMF	46.2 ± 0.01	71.6 ± 0.08	58.9 ± 0.05	49.3 ± 0.03
SS-NMF	49.1 ± 0.05	78.2 ± 0.01	62.4 ± 0.07	52.1 ± 0.04
PMM	48.8 ± 0.03	75.4 ± 0.05	61.2 ± 0.03	54.9 ± 0.06
Joint-NMF	55.7 ± 0.03	80.1 ± 0.08	65.3 ± 0.06	57.2 ± 0.09

In the following tests, different measures are introduced to evaluate the partition quality of the classical algorithms for community detection in heterogeneous networks. Since the structures of real networks are almost unknown, we adopt two standard measures widely used for clustering: normalized mutual information (NMI) [23] and clustering accuracy to quantify the partition quality of the community detection methods. For NMF-based methods, the weight parameters $\{\lambda_1, \lambda_2, \ldots, \lambda_p\}$ are set to 0.1, thus making all the bipartite graphs with the same weight. In addition, we obtain the community numbers r from the method as suggested in [4], which has been shown to well predict the number of network communities. In our experiments, we repeat each method with 50 times on all the networks and compute the average results.

The average execution times found by different algorithms are shown in Table 1. We can see that *K*means costs much less time than NMF-based algorithms, as it does not need the matrix factorization iterations. For all the real heterogeneous networks, Joint-NMF effectively accelerates the convergence speed of nonnegative matrix factorization and converges in fewer iterations and CPU seconds than other NMF methods. Because the network scales are quite different, the corresponding performances of Joint-NMF are different, too. For the larger networks, Joint-NMF has a greater competitive advantage than other methods. Our method is only slower than *K*means and NMF, which, however, has much worse clustering performance.

Tables 2 and 3, respectively, show the clustering accuracy and NMI values found by different algorithms. The methods using semisupervised learning, including SS-NMF, PMM, and Joint-NMF, generally achieve better clustering results. Therefore, we can conclude from the experiment results that multiview learning gives full consideration to the multimode/multidimension information, and it is a better choice for mining the community structure in heterogeneous networks.

Joint-NMF method attains the maximum NMI and clustering accuracy in community structure for most test cases, which means that our method has better partition quality, and achieves accuracy community structure on the real heterogeneous networks. More importantly, our method does not suffer from the problems of modularity optimization methods and makes full use of the duality information of multimode nodes, which can greatly enhance the performance of clustering algorithms. Therefore, compared to the other 4 methods, we can conclude that Joint-NMF has competitive clustering performance in terms of both accuracy and partition quality against popular community detection methods.

5. Conclusions

In this work, we introduce a multiview learning algorithm of community discovery based on nonnegative matrix factorization. In order to reveal the underlying community structure embedded in heterogeneous networks, we divide the datasets into some relational bipartite graphs and require those graphs learnt from factorizations with multiple views towards a common consensus. To achieve this, we introduce multiview learning in the heterogeneous data mining with matrix factorization and finally make all the learners reach a consensus about network partition. Moreover, we design an optimal iterative procedure to ensure the matrix factorization is simple and meaningful in terms of clustering. Through multiview learning, we are able to discover the hidden global structure in the heterogeneous networks, which seamlessly integrates multiple data types to provide us with a better picture of the underlying community distribution, highly valuable in most real-world applications.

Different form the traditional methods, our work is an instructive attempt to discover the multimode or multidimensional structure in heterogeneous networks. Actually, our Joint-NMF framework jointly takes intertype and intratype information of target nodes into considerations, thus makes the partitioning results more reasonable and effective, and detects communities with high accuracy and quality. Experimental results on four real-world datasets show that our algorithm is a competitive method to explore community structures in heterogeneous networks.

Notations

X: Heterogeneous networks dataset
X_1^p: The bipartite graph describing the relationships between mode 1 and mode p
O: Objective function a heterogeneous network
p: The count mode in heterogeneous networks
H^*: The coefficient matrix factorization from all the bipartite graphs
$Q_{r\times r}$: Auxiliary matrix for simplifying the iterative procedure
W^i: The ith basis matrix factorization from X_1^i
H^i: The ith coefficient matrix factorization from X_1^i.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Foundation of China under Grant no. 61271253.

References

[1] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 267–273, ACM, Toronto, Canada, 2003.

[2] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[4] N. P. Nguyen, T. N. Dinh, S. Tokala et al., "Overlapping communities in dynamic networks: their detection and mobile applications," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, pp. 85–96, ACM, Las Vegas, Nev, USA, 2011.

[5] D. He, D. Jin, C. Baquero, and D. Liu, "Link community detection using generative model and nonnegative matrix factorization," *PLoS ONE*, vol. 9, no. 1, Article ID e86899, 2014.

[6] Y. Chen, L. Wang, and M. Dong, "Non-Negative matrix factorization for semisupervised heterogeneous data coclustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1459–1474, 2010.

[7] L. Tang, X. Wang, and H. Liu, "Community detection in multidimensional networks," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pp. 352–359, IEEE, Athens, Greece, 2012.

[8] T. Wang, Y. Liu, and Y.-Y. Xi, "Identifying community in bipartite networks using graph regularized-based non-negative matrix factorization," *Journal of Electronics and Information Technology*, vol. 37, no. 9, pp. 2238–2245, 2015.

[9] L. Yang, W. Tao, J. Xin-Sheng, L. Caixia, and X. Mingyan, "Detecting communities in 2-mode networks via fast nonnegative matrix trifactorization," *Mathematical Problems in Engineering*, vol. 2015, Article ID 937090, 10 pages, 2015.

[10] A. Benton, R. Arora, and M. Dredze, "Learning multiview embeddings of twitter users," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 14, Berlin, Germany, 2016.

[11] E. Tsivtsivadzge, H. Borgdorff, J. van de Wijert et al., "Neighborhood co-regularized multi-view spectral clustering of microbiome data," in *Proceedings of the IAPR International Workshop on Partially Supervised Learning (PSL '13)*, pp. 80–90, Springer, Nanjing, China, 2013.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[13] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 12, pp. 556–562, 2000.

[14] P. Mandayam Comar, P.-N. Tan, and A. K. Jain, "A framework for joint community detection across multiple related networks," *Neurocomputing*, vol. 76, no. 1, pp. 93–104, 2012.

[15] Z.-C. Chang, H.-C. Chen, Y. Liu, H.-T. Yu, and R.-Y. Huang, "Community detection based on joint matrix factorization in networks with node attributes," *Acta Physica Sinica*, vol. 64, no. 21, pp. 456–465, 2015.

[16] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 126–135, August 2006.

[17] C.-J. Lin, "On the convergence of multiplicative update algorithms for non-negative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, 2007.

[18] http://www.cs.cmu.edu/~WebKB/.

[19] http://people.csail.mit.edu/jrennie/20Newsgroups/.

[20] A. MeCallum, K. Nlgam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval Journal*, vol. 3, no. 2, pp. 127–163, 2000.

[21] R. Jschke, L. Marinho, A. Hotho et al., "Tag recommendations in folksonomies," in *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD '07)*, pp. 506–514, Springer, Warsaw, Poland, September 2007.

[22] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Mining and Knowledge Discovery*, vol. 25, no. 1, pp. 1–33, 2012.

[23] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 9, pp. 219–228, 2005.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8679079, 11 pages
https://doi.org/10.1155/2017/8679079

Research Article

Games Based Study of Nonblind Confrontation

Yixian Yang,[1,2,3] Xinxin Niu,[1,2,3] and Haipeng Peng[2,3]

[1]*Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China*
[2]*Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Haipeng Peng; penghaipeng@bupt.edu.cn

Received 4 January 2017; Accepted 20 March 2017; Published 19 April 2017

Academic Editor: Liu Yuhong

Security confrontation is the second cornerstone of the General Theory of Security. And it can be divided into two categories: blind confrontation and nonblind confrontation between attackers and defenders. In this paper, we study the nonblind confrontation by some well-known games. We show the probability of winning and losing between the attackers and defenders from the perspective of channel capacity. We establish channel models and find that the attacker or the defender wining one time is equivalent to one bit transmitted successfully in the channel. This paper also gives unified solutions for all the nonblind confrontations.

1. Introduction

The core of all security issues represented by cyberspace security [1], economic security, and territorial security is confrontation. Network confrontation [2], especially in big data era [3], has been widely studied in the field of cyberspace security. There are two strategies in network confrontation: blind confrontation and nonblind confrontation. The so-called "blind confrontation" is the confrontation in which both the attacker and defender are only aware of their self-assessment results and know nothing about the enemy's assessment results after each round of confrontation. The superpower rivalry, battlefield fight, network attack and defense, espionage war, and other brutal confrontations, usually belong to the blind confrontation. The so-called "nonblind confrontation" is the confrontation in which both the attacker and defender know the consistent result after each round. The games studied in this paper are all belonging to the nonblind confrontation.

"Security meridian" is the first cornerstone of the General Theory of Security which has been well established [4, 5]. Security confrontation is the second cornerstone of the General Theory of Security, where we have studied the blind confrontation and gave the precise limitation of hacker attack ability (honker defense ability) [4, 5]. Comparing with the blind confrontation, the winning or losing rules of nonblind confrontation are more complex and not easy to study. In this paper, based on the Shannon Information Theory [6], we study several well-known games of the nonblind confrontation: "rock-paper-scissors" [7], "coin tossing" [8], "palm or back," "draw boxing," and "finger guessing" [9], from a novel point of view. The famous game, "rock-paper-scissors," has been played for thousands of years. However, there are few related analyses on it. The interdisciplinary team of Zhejiang University, Chinese Academy of Sciences, and other institutions, in cooperation with more than three hundred volunteers, spent four years playing "rock-paper-scissors" and giving corresponding analysis of game. And the findings were awarded as "Best of 2014: MIT technology review."

We obtain some significant results. The contributions of this paper are as follows:

(i) Channel models of all the above three games are established.

(ii) The conclusion that the attacker or the defender wining one time is equivalent to one bit transmitted successfully in the channel is found.

(iii) Unified solutions for all the nonblind confrontations are given.

The rest of the paper is organized as follows. The model of rock-paper-scissors is introduced in Section 2, models of coin tossing and palm or back are introduced in Section 3, models of finger guessing and drawing boxing are introduced in Section 4, unified model of linear separable nonblind confrontation is introduced in Section 5, and Section 6 concludes this paper.

2. Model of Rock-Paper-Scissors

2.1. Channel Modeling. Suppose A and B play "rock-paper-scissors," whose states can be, respectively, represented by random variables X and Y:

$X = 0$, $X = 1$, and $X = 2$ denote the "scissors," "rock," and "paper" of A, respectively;

$Y = 0$, $Y = 1$, and $Y = 2$ denote the "scissors," "rock," and "paper" of B, respectively.

Law of Large Numbers indicates that the limit of the frequency tends to probability; thus the choice habits of A and B can be represented as the probability distribution of random variables X and Y:

$\Pr(X = 0) = p$ means the probability of A for "scissors";

$\Pr(X = 1) = q$ means the probability of A for "rock";

$\Pr(X = 2) = 1 - p - q$ means the probability of A for "paper", where $0 < p, q$ and $p + q < 1$;

$\Pr(Y = 0) = r$ means the probability of B for "scissors";

$\Pr(Y = 1) = s$ means the probability of B for "rock";

$\Pr(Y = 2) = 1 - r - s$ means the probability of B for "paper," where $0 < r, s$ and $r + s < 1$.

Similarly, the joint probability distribution of two-dimensional random variables (X, Y) can be listed as follows:

$\Pr(X = 0, Y = 0) = a$ means the probability of A for "scissors" and B for "scissors";

$\Pr(X = 0, Y = 1) = b$ means the probability of A for "scissors" and B for "rock";

$\Pr(X = 0, Y = 2) = p - a - b$ means the probability of A for "scissors" and B for "paper," where $0 < a, b$, and $a + b < p$;

$\Pr(X = 1, Y = 0) = e$ means the probability of A for "rock" and B for "scissors";

$\Pr(X = 1, Y = 1) = f$ means the probability of A for "rock" and B for "rock";

$\Pr(X = 1, Y = 2) = q - e - f$ means the probability of A for "rock" and B for "paper," where $0 < e, f$, and $e + f < q$;

$\Pr(X = 2, Y = 0) = g$ means the probability of A for "paper" and B for "scissors";

$\Pr(X = 2, Y = 1) = h$ means the probability of A for "paper" and B for "rock";

$\Pr(X = 2, Y = 2) = 1 - p - q - g - h$ means the probability of A for "paper" and B for "paper," where $0 < e, f$, and $e + f < 1 - p - q$.

Construct another random variable $Z = [2(1 + X + Y)] \bmod 3$ from X and Y. Because any two random variables can form a communication channel, we get a communication channel $(X; Z)$ with X as the input and Z as the output, which is called "Channel A," which is shown in Figure 1.

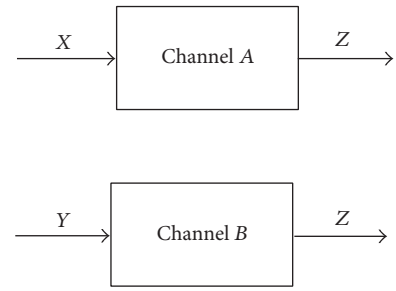


Figure 1: Block diagram of the channel model.

If A wins, then there are only three cases.

Case 1. "A chooses scissors, B chooses paper"; namely, "$X = 0, Y = 2$." This is also equivalent to "$X = 0, Z = 0$"; namely, the input of "Channel A" is equal to the output.

Case 2. "A chooses stone, B chooses scissors"; namely, "$X = 1, Y = 0$." This is also equivalent to "$X = 1, Z = 1$"; namely, the input of "Channel A" is equal to the output.

Case 3. "A chooses cloth, B chooses stone"; namely, "$X = 2, Y = 1$." This is also equivalent to "$X = 2, Z = 2$"; namely, the input of "Channel A" is equal to the output.

In contrast, if "Channel A" sends one bit from the sender to the receiver successfully, then there are only three possible cases.

Case 1. The input and the output equal 0; namely, "$X = 0, Z = 0$." This is also equivalent to "$X = 0, Y = 2$"; namely, "A chooses scissors, B chooses paper"; A wins.

Case 2. The input and the output equal 1; namely, "$X = 1, Z = 1$." This is also equivalent to "$X = 1, Y = 0$"; namely, "A chooses rock, B chooses scissors"; A wins.

Case 3. The input and the output equal 2; namely, "$X = 2, Z = 2$." This is also equivalent to "$X = 2, Y = 1$"; namely, "A chooses paper, B chooses rock"; A wins.

Based on the above six cases, we get an important lemma.

Lemma 1. *A wins once if and only if "Channel A" sends one bit from the sender to the receiver successfully.*

Now we can construct another channel $(Y; Z)$ by using random variables Y and Z with Y as the input and Z as the output, which is called "Channel B." Then similarly, we can get the following lemma.

Lemma 2. *B wins once if and only if "Channel B" sends one bit from the sender to the receiver successfully.*

Thus, the winning and losing problem of "rock-paper-scissors" played by A and B converts to the problem of whether the information bits can be transmitted successfully by "Channel A" and "Channel B." According to Shannon's second theorem [3], we know that channel capacity is equal to the maximal number of bits that the channel can transmit

successfully. Therefore, the problem is transformed into the channel capacity problem. More accurately, we have the following theorem.

Theorem 3 ("rock-paper-scissors" theorem). *If one does not consider the case that both A and B have the same state; then*

(1) *for A, there must be some skills (corresponding to the Shannon coding) for any $k/n \le C$, such that A wins k times in nC rounds of the game; if A wins u times in m rounds of the game, then $u \le mC$, where C is the capacity of "Channel A";*

(2) *for B, there must be some skills (corresponding to the Shannon coding) for any $k/n \le D$, such that B wins k times in nD rounds of the game; if B wins u times in m rounds of the game, then $u \le mD$, where D is the capacity of "Channel B";*

(3) *statistically, if $C < D$, B will win; if $C > D$, A will win; if $C = D$, A and B are evenly matched.*

Here, we calculate the channel capacity of "Channel A" and "Channel B" as follows.

For channel $(X; Z)$ of A: P denotes its transition probability matrix with $3 * 3$ order,

$$\begin{aligned}
P(0,0) &= \Pr(Z=0 \mid X=0) = \frac{(p-a-b)}{p},\\
P(0,1) &= \Pr(Z=1 \mid X=0) = \frac{b}{p},\\
P(0,2) &= \Pr(Z=2 \mid X=0) = \frac{a}{p},\\
P(1,0) &= \Pr(Z=0 \mid X=1) = \frac{f}{q},\\
P(1,1) &= \Pr(Z=1 \mid X=1) = \frac{e}{q},\\
P(1,2) &= \Pr(Z=2 \mid X=1) = \frac{(q-e-f)}{q},\\
P(2,0) &= \Pr(Z=0 \mid X=2) = \frac{g}{(1-p-q)},\\
P(2,1) &= \Pr(Z=1 \mid X=2) = \frac{(1-p-q-g-h)}{(1-p-q)},\\
P(2,2) &= \Pr(Z=2 \mid X=2) = \frac{h}{(1-p-q)}.
\end{aligned} \tag{1}$$

The channel transfer probability matrix is used to calculate the channel capacity: solve the equations $Pm = n$, where m is the column vector:

$$\begin{aligned}
m &= (m_0, m_1, m_2)^T,\\
n &= \left(\sum_{j=0}^{2} P(0,j) \log_2 P(0,j), \sum_{j=0}^{2} P(1,j) \log_2 P(1,j), \right.\\
&\quad \left. \sum_{j=0}^{2} P(2,j) \log_2 P(2,j) \right).
\end{aligned} \tag{2}$$

Consider the transition probability matrix P.

(1) If P is reversible, there is a unique solution; that is, $m = P^{-1}n$; then $C = \log_2(\sum_{j=0}^{2} 2^{m_j})$.

According to the formula $P_z(j) = 2^{m_j - C}$, $P_z(j) = \sum_{j=0}^{2} P_x(i)P(i,j)$, $i, j = 0, 1, 2$, where $P_z(j)$ is the probability distribution of Z.

And the probability distribution of X is obtained. If $P_X(i) \ge 0$, $i = 0, 1, 2$, the channel capacity can be confirmed as C.

(2) If P is irreversible, the equation has multiple solutions. Repeat the above steps; then we can get multiple C and the corresponding $P_X(i)$. If $P_X(i)$ does not satisfy $P_X(i) \ge 0$, $i = 0, 1, 2$, we delete the corresponding C.

For channel $(Y; Z)$ of B: Q denotes its transition probability matrix with $3 * 3$ order,

$$\begin{aligned}
Q(0,0) &= \Pr(Z=0 \mid Y=0) = \frac{g}{r},\\
Q(0,1) &= \Pr(Z=1 \mid Y=0) = \frac{e}{r},\\
Q(0,2) &= \Pr(Z=2 \mid Y=0) = \frac{(r-g-e)}{r},\\
Q(1,0) &= \Pr(Z=0 \mid Y=1) = \frac{f}{s},\\
Q(1,1) &= \Pr(Z=1 \mid Y=1) = \frac{b}{s},\\
Q(1,2) &= \Pr(Z=2 \mid Y=1) = \frac{(s-f-b)}{s},\\
Q(2,0) &= \Pr(Z=0 \mid Y=2) = \frac{(1-a-b)}{(1-r-s)},\\
Q(2,1) &= \Pr(Z=1 \mid Y=2) = \frac{(1-g-h)}{(1-r-s)},\\
Q(2,2) &= \Pr(Z=2 \mid Y=2) = \frac{(1-e-f)}{(1-r-s)}.
\end{aligned} \tag{3}$$

The channel transfer probability matrix Q is used to calculate the channel capacity B.

Solution equation group $Qw = u$, where w, u are the column vector:

$$\begin{aligned}
w &= (w_0, w_1, w_2)^T,\\
u &= \left(\sum_{j=0}^{2} Q(0,j) \log_2 Q(0,j), \sum_{j=0}^{2} Q(1,j) \log_2 Q(1,j), \right.\\
&\quad \left. \sum_{j=0}^{2} Q(2,j) \log_2 Q(2,j) \right).
\end{aligned} \tag{4}$$

Consider the transition probability matrix Q.

(1) If Q is reversible, there is a unique solution; that is, $w = Q^{-1}u$; then $D = \log_2(\sum_{j=0}^{2} 2^{w_j})$.

According to the formula $Q_z(j) = 2^{w_j - D}$, $Q_z(j) = \sum_{j=0}^{2} Q_y(i)Q(i,j)$, $i, j = 0, 1, 2$.

And the probability distribution of Y is obtained. If $Q_Y(i) \ge 0$, $i = 0, 1, 2$, the channel capacity can be confirmed as D.

(2) If Q is irreversible, the equation has multiple solutions. Repeat the above steps, then we can get multiple D and the corresponding $Q_Y(i)$. If $Q_Y(i)$ does not satisfy $Q_Y(i) \geq 0$, $i = 0, 1, 2$, we delete the corresponding D.

In the above analysis, the problem of "rock-paper-scissors" game has been solved perfectly, but the corresponding analysis is complex. Here, we give a more abstract and simple solution.

Law of Large Numbers indicates that the limit of the frequency tends to probability; thus the choice habits of A and B can be represented as the probability distribution of random variables X and Y:

$$\begin{aligned}
&0 < \Pr(X = x) = p_x < 1, \\
&\qquad x = 0, 1, 2, \ p_0 + p_1 + p_2 = 1; \\
&0 < \Pr(Y = y) = q_y < 1, \\
&\qquad y = 0, 1, 2, \ q_0 + q_1 + q_2 = 1; \\
&0 < \Pr(X = x, Y = y) = t_{xy} < 1, \\
&\qquad x, y = 0, 1, 2, \ \sum_{0 \leq x, y \leq 2} t_{xy} = 1;
\end{aligned} \tag{5}$$

$$p_x = \sum_{0 \leq y \leq 2} t_{xy}, \quad x = 0, 1, 2;$$

$$q_y = \sum_{0 \leq y \leq 2} t_{xy}, \quad y = 0, 1, 2.$$

The winning and losing rule of the game is if $X = x$, $Y = y$, then the necessary and sufficient condition of the winning of $A(X)$ is $(y - x) \bmod 3 = 2$.

Now construct another random variable $F = (Y - 2) \bmod 3$. Considering a channel $(X; F)$ consisting of X and F, that is, a channel with X as an input and F as an output, then, there are the following event equations.

If $A(X)$ wins in a certain round, then $(Y - X) \bmod 3 = 2$, so $F = (Y - 2) \bmod 3 = [(2 + X) - X] \bmod 3 = X$. That is, the input (X) of the channel $(X; F)$ always equals its output (F). In other words, one bit is successfully transmitted from the sender to the receiver in the channel.

Conversely, if "one bit is successfully transmitted from the sender to the receiver in the channel," it means that the input (X) of the channel $(X; F)$ always equals its output (F). That is, $F = (Y - 2) \bmod 3 = X$, which is exactly the necessary and sufficient conditions for X winning.

Based on the above discussions, $A(X)$ winning once means that the channel $(X; F)$ sends one bit from the sender to the receiver successfully and vice versa. Therefore, the channel $(X; F)$ can also play the role of "Channel A" in the third section.

Similarly, if the random variable $G = (X - 2) \bmod 3$, then the channel $(Y; G)$ can play the role of the above "Channel B."

And now the form of channel capacity for channel $(X; F)$ and channel $(Y; G)$ will be simpler. We have

$C(X; F) = \max_X [I(X, F)] = \max_X [I(X, (Y - 2) \bmod 3)] = \max_X [I(X, Y)] = \max_X [\sum t_{xy} \log(t_{xy}/(p_x q_y))]$. The maximal value here is taken for all possible t_{xy} and p_x. So, $C(X; F)$ is actually the function of q_0, q_1, and q_2.

Similarly, $C(Y; G) = \max_Y [I(Y, G)] = \max_Y [I(Y, (X - 2) \bmod 3)] = \max_Y [\mathrm{I}(X, Y)] = \max_Y [\sum t_{xy} \log(t_{xy}/(p_x q_y))]$. The maximal value here is taken for all possible t_{xy} and q_y. So, $C(Y; G)$ is actually the function of p_0, p_1, and p_2.

2.2. The Strategy of Win. According to Theorem 3, if the probability of a specific action is determined, the victory of both parties in the "rock-paper-scissors" game is determined as well. In order to obtain the victory with higher probability, one must adjust his strategy.

2.2.1. The Game between Two Fools. The so-called "two fools" means that A and B entrench their habits; that is, they choose their actions in accordance with the established habits no matter who won in the past. According to Theorem 3, statistically, if $C < D$, A will lose; if $C > D$, then A will win; and if $C = D$, then both parties are well-matched in strength.

2.2.2. The Game between a Fool and a Sage. If A is a fool, he still insists on his inherent habit; then after confronting a sufficient number of times, B can calculate the distribution probabilities p and q of random variable X corresponding to A. And B can get the channel capacity of A by some related conditional probability distribution at last, and then by adjusting their own habits (i.e., the probability distribution of the random variable Y and the corresponding conditional probability distribution, etc.); then B enlarges his own channel capacity to make the rest of game more beneficial to himself; moreover, the channel capacity of B is larger enough, $C(B) > C(A)$; then B win the success at last.

2.2.3. The Game between Two Sages. If both A and B get used to summarizing the habits of each other at any time, and adjust their habits, enlarge their channel capacity. At last, the two parties can get the equal value of channel capacities; that is, the competition between them will tend to a balance, a dynamically stable state.

3. Models of "Coin Tossing" and "Palm or Back"

3.1. The Channel Capacity of "Coin Tossing" Game. "Coin tossing" game: "banker" covers a coin under his hand on the table, and "player" guesses the head or tail of the coin. The "player" will win when he guesses correctly.

Obviously, this game is a kind of "nonblind confrontation." We will use the method of channel capacity to analyze the winning or losing of the game.

Based on the Law of Large Numbers in the probability theory, the frequency tends to probability. Thus, according to the habits of "banker" and "player," that is, the statistical regularities of their actions in the past, we can give the probability distribution of their actions.

We use the random variable X to denote the state of the "banker." $X = 0$ $(X = 1)$ means the coin is head (tail). So the habit of "banker" can be described by the probability distribution of X; that is, $\Pr(X = 0) = p$, $\Pr(X = 1) = 1 - p$, where $0 \leq p \leq 1$.

We use the random variable Y to denote the state of the "player." $Y = 0$ $(Y = 1)$ means that he guesses head (tail).

So the habit of "player" can be described by the probability distribution of Y; that is, $\Pr(Y = 0) = q$, $\Pr(Y = 1) = 1 - q$, where $0 \le q \le 1$. Similarly, according to the past states of "banker" and "player," we have the joint probability distribution of random variables (X, Y); namely,

$$\begin{aligned} &\Pr(X = 0, Y = 0) = a; \\ &\Pr(X = 0, Y = 1) = b; \\ &\Pr(X = 1, Y = 0) = c; \\ &\Pr(X = 1, Y = 1) = d, \end{aligned} \tag{6}$$

where $0 \le p, q, a, b, c, d \le 1$ and

$$\begin{aligned} &a + b + c + d = 1; \\ &p = \Pr(X = 0) \\ &\quad = \Pr(X = 0, Y = 0) + \Pr(X = 0, Y = 1) \\ &\quad = a + b; \\ &q = \Pr(Y = 0) \\ &\quad = \Pr(X = 0, Y = 0) + \Pr(X = 1, Y = 0) \\ &\quad = a + c. \end{aligned} \tag{7}$$

Taking X as the input and Y as the output, we obtain the channel $(X; Y)$ which is called "Channel X" in this paper.

Because Y guesses correctly $= \{X = 0, Y = 0\} \cup \{X = 1, Y = 1\}$ = one bit is successfully transmitted from the sender X to the receiver Y in "Channel X," "Y wins one time" is equivalent to transmitting one bit of information successfully in "Channel X."

Based on the channel coding theorem of Shannon's Information Theory, if the capacity of "Channel X" is C, for any transmission rate $k/n \le C$, we can receive k bits successfully by sending n bits with an arbitrarily small probability of decoding error. Conversely, if "Channel X" can transmit s bits to the receiver by sending n bits without error, there must be $S \le nC$. In a word, we have the following theorem.

Theorem 4 (banker theorem). *Suppose that the channel capacity of "Channel X" composed of the random variable $(X; Y)$ is C. Then one has the following: (1) if Y wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close to 1 in the k/C rounds; conversely, (2) if Y wins S times in n rounds, there must be $S \le nC$.*

According to Theorem 3, we only need to figure out the channel capacity C of "Channel X"; then the limitation of times that "Y wins" is determined. So we can calculate the transition probability matrix $A = [A(i, j)]$, $i, j = 0, 1$ of "Channel X":

$$\begin{aligned} A(0,0) &= \Pr(Y = 0 \mid X = 0) = \frac{\Pr(Y = 0, X = 0)}{\Pr(X = 0)} \\ &= \frac{a}{p}; \\ A(0,1) &= \Pr(Y = 1 \mid X = 0) = \frac{\Pr(Y = 1, X = 0)}{\Pr(X = 0)} \\ &= \frac{b}{p} = 1 - \frac{a}{p}; \\ A(1,0) &= \Pr(Y = 0 \mid X = 1) = \frac{\Pr(Y = 0, X = 1)}{\Pr(X = 1)} \\ &= \frac{c}{(1-p)} = \frac{(q-a)}{(1-p)}; \\ A(1,1) &= \Pr(Y = 1 \mid X = 1) = \frac{\Pr(Y = 1, X = 1)}{\Pr(X = 1)} \\ &= \frac{d}{(1-p)} = 1 - \frac{(q-a)}{(1-p)}. \end{aligned} \tag{8}$$

Thus, the mutual information $I(X, Y)$ of X and Y equals

$$\begin{aligned} I(X,Y) &= \sum_X \sum_Y p(X,Y) \log\left(\frac{p(X,Y)}{[p(X)\,p(Y)]}\right) \\ &= a \log\left[\frac{a}{(pq)}\right] + b \log\left[\frac{b}{[p(1-q)]}\right] \\ &\quad + c \log\left[\frac{c}{[(1-p)q]}\right] \\ &\quad + d \log\left[\frac{d}{[(1-p)(1-q)]}\right] \\ &= a \log\left[\frac{a}{(pq)}\right] + (p-a) \log\left[\frac{(p-a)}{[p(1-q)]}\right] \\ &\quad + (q-a) \log\left[\frac{(q-a)}{[(1-p)q]}\right] \\ &\quad + (1+a-p-q) \log\left[\frac{(1+a-p-q)}{[(1-p)(1-q)]}\right]. \end{aligned} \tag{9}$$

Thus, the channel capacity C of "Channel X" is equal to $\max[I(X, Y)]$ (the maximal value here is taken from all possible binary random variables X). In a word, $C = \max[I(X, Y)]$ $0 < a, p < 1$ (where $I(X, Y)$ is the mutual information above). Thus, the channel capacity C of "Channel X" is a function of q, which is defined as $C(q)$.

Suppose the random variable $Z = (X + 1) \bmod 2$. Taking Y as the input and Z as the output, we obtain the channel $(Y; Z)$ which is called "Channel Y" in this paper.

Because $\{X \text{ wins}\} = \{Y = 0, X = 1\} \cup \{Y = 1, X = 0\} = \{Y = 0, Z = 0\} \cup \{Y = 1, Z = 1\} = \{$one bit is successfully transmitted from the sender Y to the receiver Z in the "Channel Y"$\}$, "X wins one time" is equivalent to transmitting one bit of information successfully in "Channel Y."

Based on the Channel coding theorem of Shannon's Information Theory, if the capacity of "Channel Y" is D, for any transmission rate $k/n \le D$, we can receive k bits successfully by sending n bits with an arbitrarily small probability of

decoding error. Conversely, if "Channel Y" can transmit s bits to the receiver by sending n bits without error, there must be $S \le nD$. In a word, we have the following theorem.

Theorem 5 (player theorem). *Suppose that the channel capacity of "Channel Y" composed of the random variable $(Y; Z)$ is D. Then one has the following: (1) if X wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close to 1 in the k/C rounds; conversely, (2) if X wins S times in the n rounds, there must be $S \le nD$.*

According to Theorem 4, we can determine the winning limitation of X as long as we know the channel capacity D of "Channel Y."

Similarly, we can get the channel capacity $D = \max[I(Y, Z)]$, $0 < a, q < 1$, of "Channel Y." Thus, the channel capacity D of "Channel Y" is a function of p, which is denoted as $D(p)$.

$$\begin{aligned} I(Y,Z) &= \sum_{Y}\sum_{Z} p(Y,Z)\log\left(\frac{p(Y,Z)}{[p(Y)\,p(Z)]}\right) \\ &= a\log\left[\frac{a}{(pq)}\right] + (p-a)\log\left[\frac{(p-a)}{[p(1-q)]}\right] \\ &\quad + (q-a)\log\left[\frac{(q-a)}{[(1-p)q]}\right] \\ &\quad + (1+a-p-q)\log\left[\frac{(1+a-p-q)}{[(1-p)(1-q)]}\right]. \end{aligned} \tag{10}$$

From Theorems 3 and 4, we can obtain the quantitative results of "the statistical results of winning and losing" and "the game skills of banker and player."

Theorem 6 (strength theorem). *In the game of "coin tossing," if the channel capacities of "Channel X" and "Channel Y" are $C(q)$ and $D(p)$, respectively, one has the following.*

Case 1. If both X and Y do not try to adjust their habits in the process of game, that is, p and q are constant, statistically, if $C(q) > D(p)$, Y will win; if $C(q) < D(p)$, X will win; and if $C(q) = D(p)$, the final result of the game is a "draw."

Case 2. If X implicitly adjusts his habit and Y does not, that is, change the probability distribution p of the random variable X to enlarge the $D(p)$ of "Channel Y" such that $D(p) > C(p)$, statistically, X will win. On the contrary, if Y implicitly adjusts his habit and X does not, that is, $D(p) < C(p)$, Y will win.

Case 3. If both X and Y continuously adjust their habits and make $C(q)$ and $D(p)$ grow simultaneously, they will achieve a dynamic balance when $p = q = 0.5$, and there is no winner or loser in this case.

3.2. The Channel Capacity of "Palm or Back" Game. The "palm or back" game: three participants (A, B, and C) choose their actions of "palm" or "back" at the same time; if one of the participants choose the opposite action to the others (e.g., the others choose "palm" when he chooses "back"), he will win this round.

Obviously, this game is also a kind of "nonblind confrontation." We will use the method of channel capacity to analyze the winning or losing of the game.

Based on the Law of Large Numbers in the probability theory, the frequency tends to probability. Thus, according to the habits of A, B, and C, that is, the statistical regularities of their actions in the past, we have the probability distribution of their actions.

We use the random variable X to denote the state of A. $X = 0$ ($Y = 1$) means that he chooses "palm (back)." Thus, the habit of A can be described as the probability distribution of X; that is, $\Pr(X = 0) = p$, $\Pr(X = 1) = 1 - p$, where $0 \le p \le 1$.

We use random variable Y to denote the state of B. $Y = 0$ ($Y = 1$) means that he chooses "palm (back)". Thus, the habit of B can be described as the probability distribution of X, that is, $\Pr(Y = 0) = q$, $\Pr(Y = 1) = 1 - q$, where $0 \le q \le 1$.

We use the random variable Z to denote the state of C. $Z = 0$ ($Z = 1$) means that he chooses "palm (back)." Thus, the habit of C can be described as the probability distribution of Z; that is, $\Pr(Z = 0) = r$, $\Pr(Z = 1) = 1 - r$, where $0 \le r \le 1$.

Similarly, according to the Law of Large Numbers, we can obtain the joint probability distributions of the random variables (X, Y, Z) from the records of their game results after some rounds; namely,

$$\begin{aligned} &\Pr(A \text{ for palm}, B \text{ for palm}, C \text{ for palm}) \\ &\quad = \Pr(X = 0\ Y = 0\ Z = 0) = a; \\ &\Pr(A \text{ for palm}, B \text{ for palm}, C \text{ for back}) \\ &\quad = \Pr(X = 0\ Y = 0\ Z = 1) = b; \\ &\Pr(A \text{ for palm}, B \text{ for back}, C \text{ for palm}) \\ &\quad = \Pr(X = 0\ Y = 1\ Z = 0) = c; \\ &\Pr(A \text{ for palm}, B \text{ for back}, C \text{ for back}) \\ &\quad = \Pr(X = 0\ Y = 1\ Z = 1) = d; \\ &\Pr(A \text{ for back}, B \text{ for palm}, C \text{ for palm}) \\ &\quad = \Pr(X = 1\ Y = 0\ Z = 0) = e; \\ &\Pr(A \text{ for back}, B \text{ for palm}, C \text{ for back}) \\ &\quad = \Pr(X = 1\ Y = 0\ Z = 1) = f; \\ &\Pr(A \text{ for back}, B \text{ for back}, C \text{ for palm}) \\ &\quad = \Pr(X = 1\ Y = 1\ Z = 0) = g; \\ &\Pr(A \text{ for back}, B \text{ for back}, C \text{ for back}) \\ &\quad = \Pr(X = 1\ Y = 1\ Z = 1) = h, \end{aligned} \tag{11}$$

where $0 \le p, q, r, a, b, c, d, e, f, g, h \le 1$ and

$$\begin{aligned} &a + b + c + d + e + f + g + h = 1; \\ &p = \Pr(A \text{ for palm}) = \Pr(X = 0) = a + b + c + d; \\ &q = \Pr(B \text{ for palm}) = \Pr(Y = 0) = a + b + e + f; \\ &r = \Pr(C \text{ for palm}) = \Pr(Z = 0) = a + c + e + g. \end{aligned} \tag{12}$$

Suppose the random variable $M = (X + Y + Z) \bmod 2$; then the probability distribution of M is

$$\begin{aligned}
\Pr(M=0) &= \Pr(X=0,Y=0,Z=0) \\
&\quad + \Pr(X=0,Y=1,Z=1) \\
&\quad + \Pr(X=1,Y=1,Z=0) \\
&\quad + \Pr(X=1,Y=0,Z=1) \\
&= a+d+g+f, \\
\Pr(M=1) &= \Pr(X=0,Y=0,Z=1) \\
&\quad + \Pr(X=0,Y=1,Z=0) \\
&\quad + \Pr(X=1,Y=0,Z=0) \\
&\quad + \Pr(X=1,Y=1,Z=1) \\
&= b+c+e+h.
\end{aligned} \tag{13}$$

Taking X as the input and M as the output, we obtain the channel $(X; M)$ which is called "Channel A" in this paper.

After removing the situations in which three participants choose the same actions, we have the following equation:

$\{A \text{ wins}\} = \{A \text{ for palm}, B \text{ for back}, C \text{ for back}\} \cup \{A \text{ for back}, B \text{ for palm}, C \text{ for palm}\} = \{X=0, Y=1, Z=1\} \cup \{X=1, Y=0, Z=0\} = \{X=0, M=0\} \cup \{X=1, M=1\}$ = {one bit is successfully transmitted from the sender (X) to the receiver (M) in the "Channel A"}.

Conversely, after removing the situations that three participants choose the same actions, if {one bit is successfully transmitted from sender (X) to the receiver (M) in the "Channel A"}, there is $\{X=0, M=0\} \cup \{X=1, M=1\} = \{X=0, Y=1, Z=1\} \cup \{X=1, Y=0, Z=0\} = \{A \text{ for palm}, B \text{ for back}, C \text{ for back}\} \cup \{A \text{ for back}, B \text{ for palm}, C \text{ for palm}\} = \{A \text{ wins}\}$. Thus, "$A$ wins one time" is equivalent to transmitting one bit successfully from the sender X to the receiver M in the "Channel A." From the channel coding theorem of Shannon's Information Theory, if the capacity of the "Channel A" is E, for any transmission rate $k/n \le E$, we can receive k bits successfully by sending n bits with an arbitrarily small probability of decoding error. Conversely, if the "Channel A" can transmit s bits to the receiver by sending n bits without error, there must be $S \le nE$. In a word, we have the following theorem.

Theorem 7. *Suppose that the channel capacity of the "Channel A" composed of the random variable $(X; M)$ is E. Then, after removing the situations in which three participants choose the same actions, one has the following: (1) if A wants to win k times, he must have a certain skill (corresponding to the Shannon coding theory) to achieve the goal by any probability close to 1 in the k/E rounds; conversely, (2) if A wins S times in the n rounds, there must be $S \le nE$.*

In order to calculate the channel capacity of the channel $(X; M)$, we should first calculate the joint probability distribution of the random variable (X, M):

$$\begin{aligned}
\Pr(X=0,M=0) &= \Pr(X=0,Y=0,Z=0) \\
&\quad + \Pr(X=0,Y=1,Z=1) \\
&= a+d; \\
\Pr(X=0,M=1) &= \Pr(X=0,Y=1,Z=0) \\
&\quad + \Pr(X=0,Y=0,Z=1) \\
&= c+b; \\
\Pr(X=1,M=0) &= \Pr(X=1,Y=1,Z=0) \\
&\quad + \Pr(X=1,Y=0,Z=1) \\
&= g+f; \\
\Pr(X=1,M=1) &= \Pr(X=1,Y=0,Z=0) \\
&\quad + \Pr(X=1,Y=1,Z=1) \\
&= e+h.
\end{aligned} \tag{14}$$

Therefore, the mutual information between X and M equals

$$\begin{aligned}
I(X,M) &= (a+d)\log\left[\frac{(a+d)}{[p(a+d+g+f)]}\right] \\
&\quad + (g+f)\log\left[\frac{(g+f)}{[(1-p)(a+d+g+f)]}\right] \\
&\quad + (c+b)\log\left[\frac{(c+b)}{[p(b+c+e+h)]}\right] + (e+h) \\
&\quad \cdot \log\left[\frac{(e+h)}{[(1-p)(b+c+e+h)]}\right] = (a+d) \\
&\quad \cdot \log\left[\frac{(a+d)}{[p(a+d+g+f)]}\right] + (g+f) \\
&\quad \cdot \log\left[\frac{(g+f)}{[(1-p)(a+d+g+f)]}\right] + (p-a-d) \\
&\quad \cdot \log\left[\frac{(p-a-d)}{[p(1-(a+d+f+g))]}\right] \\
&\quad + (1-(p+f+g)) \\
&\quad \cdot \log\left[\frac{(1-(p+f+g))}{[(1-p)(1-(a+d+f+g))]}\right].
\end{aligned} \tag{15}$$

Thus, the channel capacity of "channel A" is equal to $E = \max[I(X, M)]$ and it is a function of q and r, which is defined as $E(q, r)$.

Taking Y as the input and M as the output, we obtain the channel (Y, M) which is called "Channel B." Similarly, we have the following.

Theorem 8. *Suppose that the channel capacity of the "Channel B" composed of the random variable $(Y; M)$ is F. Then, after removing the situation in which the three participants choose the same action, one has the following: (1) if B wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close*

to 1 in the k/F rounds; conversely, (2) if B wins S times in the n rounds, there must be S ≤ nF.

The channel capacity F can be calculated as the same way of calculating E. Here, the capacity of "Channel B" is a function of p and r, which can be defined as $F(p, r)$.

Similarly, taking Z as the input and M as the output, we obtain the channel (Z, M) which is called "Channel C." So we have the following.

Theorem 9. *Suppose that the channel capacity of the "Channel C" composed of the random variable $(Z; M)$ is G. Then, after removing the situations in which three participants choose the same actions, one has the following: (1) if C wants to win k times, he must have a certain skill (corresponding to the Shannon coding theory) to achieve the goal by any probability close to 1 in the k/F rounds; conversely, (2) if C wins S times in the n rounds, there must be S ≤ nG.*

The channel capacity G can be calculated by the same way of calculating E. Now the capacity of "Channel C" is a function of p and q, which can be defined as $G(p, q)$.

From Theorems 6, 7, and 8, we can qualitatively describe the winning or losing situations of A, B, and C in the palm or back game.

Theorem 10. *If the channel capacities of "Channel A," "Channel B," and "Channel C" are E, F, and G, respectively, the statistical results of winning or losing depend on the values of E, F, and G. The one who has the largest channel capacity will gain the priority. We can know that the three channel capacities cannot be adjusted only by one participant individually. It is difficult to change the final results by adjusting one's habit (namely, only change one of the p, q, and r separately), unless two of them cooperate secretly.*

4. Models of "Finger Guessing" and "Draw Boxing"

4.1. Model of "Finger Guessing". "Finger guessing" is a game between the host and guest in the banquet. The rules of the game are the following. The host and the guest, respectively, choose one of the following four gestures at the same time in a round: bug, rooster, tiger, and stick. Then they decide the winner by the following regulations: "Bug" is inferior to "rooster"; "rooster" is inferior to "tiger"; "tiger" is inferior to "stick"; and "stick" is inferior to "bug". Beyond that, the game is ended in a draw and nobody will be punished.

The "host A" and "guest B" will play the "finger guessing game" again after the complete end of this round. The mathematical expression of "finger guessing game" is as follows: suppose A and B are denoted by random variables X and Y, respectively; there are 4 possible values of them. Specifically,

$X = 0$ (or $Y = 0$) when A (or B) shows "bug";

$X = 1$ (or $Y = 1$) when A (or B) shows "cock";

$X = 2$ (or $Y = 2$) when A (or B) shows "tiger";

$X = 3$ (or $Y = 3$) when A (or B) shows "stick".

If A shows x (namely, $X = x$, $0 \le x \le 3$) and B shows y (namely, $Y = y$, $0 \le y \le 3$) in a round, the necessary and sufficient condition of A wins in this round is $(x - y) \bmod 4 = 1$. The necessary and sufficient condition of B wins in this round is $(y - x) \bmod 4 = 1$. Otherwise, this round ends in a draw and proceeds to the next round of the game.

Obviously, the "finger guessing" game is a kind of "non-blind confrontation." Who is the winner and how many times the winner wins? How can they make themselves win more? We will use the "channel capacity method" of the "General Theory of Security" to answer these questions.

Based on the Law of Large Numbers in the probability theory, the frequency tends to probability. Thus, according to the habits of "host (X)" and "guest (Y)," that is, the statistical regularities of their actions in the past (if they meet for the first time, we can require them to play a "warm-up game" and record their habits), we can give the probability distribution of X, Y and the joint probability distribution of (X, Y), respectively:

$$\begin{aligned}
&0 < \Pr(X = i) = p_i < 1, \\
&\qquad i = 0, 1, 2, 3;\ p_0 + p_1 + p_2 + p_3 = 1; \\
&0 < \Pr(Y = i) = q_i < 1, \\
&\qquad i = 0, 1, 2, 3;\ q_0 + q_1 + q_2 + q_3 = 1; \\
&0 < \Pr(X = i, Y = j) = t_{ij} < 1, \\
&\qquad i, j = 0, 1, 2, 3;\ \sum_{0 \le i, j \le 3} t_{ij} = 1.
\end{aligned} \tag{16}$$

$$p_x = \sum_{0 \le y \le 3} t_{xy}, \quad x = 0, 1, 2, 3;$$

$$q_y = \sum_{0 \le x \le 3} t_{xy}, \quad y = 0, 1, 2, 3.$$

In order to analyze the winning situation of A, we construct a random variable $Z = (Y + 1) \bmod 4$. Then we use the random variables X and Z to form a channel $(X; Z)$, which is called "channel A"; namely, the channel takes X as the input and Z as the output. Then we analyze some equations of the events. If A shows x (namely, $X = x$, $0 \le x \le 3$) and B shows y (namely, $Y = y$, $0 \le y \le 3$) in a round, one has the following.

If A wins in this round, there is $(x - y) \bmod 4 = 1$; that is, $y = (x - 1) \bmod 4$, so we have $z = (y + 1) \bmod 4 = [(x - 1) + 1] \bmod 4 = x \bmod 4 = x$. In other words, the output Z is always equal to the input X in the channel A at this time. That is, a "bit" is successfully transmitted from the input X to its output Z.

In contrast, if a "bit" is successfully transmitted from the input X to the output Z in the "channel A," "the output z is always equal to its input x; namely, $z = x$" is true at this time. Then there is $(x - y) \bmod 4 = (z - y) \bmod 4 = [(y + 1) - y] \bmod 4 = 1 \bmod 4 = 1$. Hence, we can judge that "A wins" according to the rules of this game.

Combining with the situations above, one has the following.

Lemma 11. *In the "finger guessing" game, "A wins one time" is equivalent to "a 'bit' is successfully transmitted from the input to its output in the 'channel A.'" Combine Lemma 1 with the "channel coding theorem" of Shannon's Information Theory; if the capacity of the "channel A" is C, for any transmission rate k/n ≤ C, we can receive k bits successfully by sending n bits with*

an arbitrarily small probability of decoding error. Conversely, if the "channel A" can transmit s bits to the receiver by sending n bits without error, there must be $S \le nC$. In a word, we have the following theorem.

Theorem 12. *Suppose that the channel capacity of the "channel A" composed of the random variable $(X; Z)$ is C. Then after removing the situation of "draw," one has the following: (1) if A wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close to 1 in the k/C rounds; conversely, (2) if A wins S times in the n rounds, there must be $S \le nC$.*

According to Theorem 12, we only need to figure out the channel capacity C of the "channel A," then the limitation of the times of "A wins" is determined. So we can calculate the channel capacity C: first, the joint probability distribution of (X, Z) is $\Pr(X = i, Z = j) = \Pr(X = i, (Y + 1) \bmod 4 = j) = \Pr(X = i, Y = (j - 1) \bmod 4) = t_{i(j-1) \bmod 4}$, $i, j = 0, 1, 2, 3, 4$.

Therefore, the channel capacity of the channel $A(X; Z)$ is

$$C = \max [I(X, Z)] = \max \left\{ \sum_{0 \le i,j \le 3} \left[t_{i(j-1) \bmod 4} \right] \frac{\log \left[t_{i(j-1) \bmod 4} \right]}{\left(p_i q_j \right)} \right\}. \quad (17)$$

The max in the equation is the maximal value taken from the real numbers which satisfy the following conditions: $0 < p_i, t_{ij} < 1, i, j = 0, 1, 2, 3$; $p_0 + p_1 + p_2 + p_3 = 1$; $\sum_{0 \le i,j \le 3} t_{ij} = 1$; $p_x = \sum_{0 \le y \le 3} t_{xy}$. Thus, the capacity C is actually the function of the positive real variables which satisfy the following conditions $q_0 + q_1 + q_2 + q_3 = 1$ and $0 < q_i < 1$, $i = 0, 1, 2, 3$; namely, it can be written as $C(q_0, q_1, q_2, q_3)$, where $q_0 + q_1 + q_2 + q_3 = 1$.

Similarly, we can analyze the situation of "B wins." We can see that the times of "A wins" $(C(q_0, q_1, q_2, q_3))$ depend on the habit of $B(q_0, q_1, q_2, q_3)$. If both A and B stick to their habits, their winning or losing situation is determined; if either A or B adjusts his habit, he can win statistically when his channel capacity is larger; if both A and B adjust their habits, their situations will eventually reach a dynamic balance.

4.2. Model of "Draw Boxing". "Draw boxing" is more complicated than "finger guessing," and it is also a game between the host and guest in the banquet. The rule of the game is that the host (A) and the guest (B) independently show one of the six gestures from 0 to 5 and shout one of eleven numbers from 0 to 10. That is, in each round, "host A" is a two-dimensional random variable $A = (X, Y)$, where $0 \le X \le 5$ is the gesture showed by the "host" and $0 \le Y \le 10$ is the number shouted by him. Similarly, "guest B" is also a two-dimensional random variable $B = (F, G)$, where $0 \le F \le 5$ is the gesture showed by the "guest" and $0 \le G \le 10$ is the number shouted by him. If A and B are denoted by (x, y) and (f, g) in a certain round, respectively, the rules of the "draw boxing" game are

If $x + f = y$, A wins;

If $x + f = g$, B wins.

If the above two cases do not occur, the result of this round is a "draw," and A and B continue the next round. Specifically, when the numbers shouted by both sides are the same (namely, $g = y$), the result of this round is a "draw." However, the numbers shouted by both sides are different and the gestures showed by them are not equal to "the number shouted by any side"; the result of this round also comes to a "draw."

Obviously, the "draw boxing" game is a kind of "nonblind confrontation." Who is the winner and how many times the winner wins? How can they make themselves win more? We will use the channel capacity method of the "General Theory of Security" to answer these questions.

Based on the Law of Large Numbers in the probability theory, the frequency tends to probability. Thus, according to the habits of "host (A)" and "guest (B)", that is the statistical regularities of their actions in the past (if they meet for the first time, we can require them to play a "warm-up game" and record their habits), we can give the probability distribution of A, B and their components X, Y, F, and G, and the joint probability distribution of (X, Y, F, G), respectively.

The probability of "A shows x":

$0 < \Pr(X = x) = p_x < 1, 0 \le x \le 5; x_0 + x_1 + x_2 + x_3 + x_4 + x_5 = 1;$

The probability of "B shows f":

$0 < \Pr(F = f) = q_f < 1, 0 \le f \le 5; f_0 + f_1 + f_2 + f_3 + f_4 + f_5 = 1;$

The probability of "A shouts y":

$0 < \Pr(Y = y) = r_y < 1, 0 \le y \le 10; \sum_{0 \le y \le 10} r_y = 1;$

The probability of "B shouts g":

$0 < \Pr(G = g) = s_g < 1, 0 \le g \le 10; \sum_{0 \le g \le 10} s_g = 1;$

The probability of "A shows x and shouts y":

$0 < \Pr[A = (x, y)] = \Pr(X = x, Y = y) = b_{xy} < 1, 0 \le y \le 10, 0 \le x \le 5, \sum_{0 \le y \le 10, 0 \le x \le 5} b_{xy} = 1;$

The probability of "B shows f and shouts g":

$0 < \Pr[B = (f, g)] = \Pr(F = f, G = g) = h_{fg} < 1, 0 \le g \le 10, 0 \le f \le 5, \sum_{0 \le g \le 10, 0 \le f \le 5} h_{fg} = 1;$

The probability of "A shows x and shouts y" and "B shows f and shouts g" at the same time:

$0 < \Pr[A = (x, y), B = (f, g)] = \Pr(X = x, Y = y, F = f, G = g) = t_{xyfg} < 1$, where $0 \le y, g \le 10$, $0 \le x, f \le 5$, $\sum_{0 \le y, g \le 10, 0 \le x, f \le 5} t_{xyfg} = 1$.

In order to analyze the situation of A wins, we construct a two-dimensional random variable

$$Z = (U, V) = (X\delta(G - Y), X + F). \quad (18)$$

The function δ is defined as $\delta(0) = 0$; $\delta(x) = 1$ when $x \ne 0$. Therefore,

$$\Pr[Z = (u, v)] = \sum_{x+f=v, x\delta(g-y)=u} t_{xyfg} =: d_{uv}, \quad \text{where } 0 \le v \le 10,\ 0 \le u \le 5. \quad (19)$$

Then, we use the random variables A and Z to form a channel $(A; Z)$, which is called "channel A" and takes A as the input and Z as the output.

Then we analyze some equations. In a certain round, A shows x (i.e., $X = x, 0 \le x \le 5$) and shouts y (i.e., $Y = y, 0 \le y \le 10$); meanwhile, B shows f (i.e., $F = f, 0 \le f \le 5$) and

shouts g (i.e., $G = g, 0 \le g \le 10$). According to the evaluation rules, we have the following: if A wins in this around, we have $x + f = y$ and $y \ne g$. Thus, $\delta(g - y) = 1$ and $Z = (u, v) = (x\delta(g - y), x + f) = (x, y) = A$. In other words, the output Z of the "channel A" is always equal to its input A at this time; that is to say, a "bit" is sent successfully from the input A to its output Z.

In contrast, if one bit is successfully sent from the input A to the output Z in the "channel A," "the output $z = (u, v) = (x\delta(g - y)x + f)$" is always equal to the "input (x, y)" at this time; also there is $x\delta(g - y) = x$ when $x + f = y$; that is, $y \ne g$ and $x + f = y$. Thus, A wins this round according to the evaluation rules.

Combining with the cases above, we have the following.

Lemma 13. *In a "draw boxing" game, "A wins one time" is equivalent to one bit is successfully sent from the input of "channel A" to its output.*

Combining Lemma 13 with the "channel coding theorem" of Shannon's Information Theory, if the capacity of the "channel A" is D, for any transmission rate $k/n \le D$, we can receive k bits successfully by sending n bits with an arbitrarily small probability of decoding error. Conversely, if the "channel A" can transmit s bits to the receiver by sending n bits without error, there must be $S \le nD$. In a word, we have the following theorem.

Theorem 14. *Suppose that the channel capacity of the "channel A" composed of the random variable $(A; Z)$ is D. Then after removing the situation of "draw," one has the following: (1) if A wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close to 1 in the k/D rounds; conversely, (2) if A wins S times in the n rounds, there must be $S \le nD$.*

According to Theorem 4, we only need to figure out the channel capacity D of the "channel A"; then the limitation of times that "A wins" is determined. So we can calculate the channel capacity D:

$$\begin{aligned} D &= \max\left[I(A, Z)\right] = \max\left\{\sum_{a,z} \Pr(a, z) \cdot \log\left[\frac{\Pr(a, z)}{\left[\Pr(a)\Pr(z)\right]}\right]\right\} \\ &= \max\left\{\sum_{x,y,f,g} \Pr\left(x, y, x\delta\left(g - y\right), x + f\right) \cdot \log\left[\frac{\Pr\left(x, y, x\delta\left(g - y\right), x + f\right)}{\left[\Pr\left(x, y\right)\Pr\left(x\delta\left(g - y\right), x + f\right)\right]}\right]\right\} \\ &= \max\left\{\sum_{x,y,f,g} t_{x,y,x\delta(g-y),x+f} \cdot \log\left[\frac{t_{x,y,x\delta(g-y),x+f}}{\left[b_{xy}d_{x\delta(g-y),x+f}\right]}\right]\right\}. \end{aligned} \tag{20}$$

The maximal value in the equation is a real number which satisfies the following conditions:

$$\begin{aligned} &\sum_{0\le y\le 10} r_y = 1; \quad 0 \le y \le 10; \\ &\sum_{0\le y\le 10, 0\le x\le 5} b_{xy} = 1; \quad 0 \le y \le 10,\ 0 \le x \le 5, \\ &\sum_{0\le g\le 10, 0\le f\le 5} h_{fg} = 1; \quad 0 \le g \le 10,\ 0 \le f \le 5. \end{aligned} \tag{21}$$

Thus, the capacity D is actually the function of f_i, g_j, which satisfies the following conditions: $0 \le f \le 5$; $f_0 + f_1 + f_2 + f_3 + f_4 + f_5 = 1$; $0 \le g \le 10$; $\sum_{0\le g\le 10} s_g = 1$, where $0 \le i \le 5$ and $0 \le j \le 10$.

Similarly, we can analyze the situation of "B wins." We can see that the times of "A wins" $(D(g_j, f_i))$ depend on the habit of $B(g_j, f_i)$. If both A and B stick to their habits, their winning or losing is determined; if either A or B adjusts his habit, he can win statistically when his channel capacity is larger; if both A and B adjust their habits, their situations will eventually reach a dynamic balance.

5. Unified Model of Linear Separable "Nonblind Confrontation"

Suppose that the hacker (X) has n methods of attack; that is, the random variable X has n values which can be denoted as $\{x_0, x_1, \ldots, x_{n-1}\} = \{0, 1, 2, \ldots, n-1\}$. These n methods make up the entire "arsenal" of the hacker.

Suppose that the honker (Y) has m methods of defense; that is, the random variable Y has m values, which can be denoted as $\{y_0, y_1, y_{m-1}\} = \{0, 1, 2, \ldots, m-1\}$. These m methods make up the entire "arsenal" of the honker.

Remark 15. In the following, we will equivalently transform between "the methods x_i, y_j" and "the numbers i, j" as needed; that is, $x_i = i$ and $y_j = j$. By the transformation, we can make the problem clear in the interpretation and simple in the form.

In the nonblind confrontation, there is a rule of winning or losing between each hacker's method x_i $(i = 0, 1, \ldots, n-1)$ and each honker's method y_j $(j = 0, 1, \ldots, m-1)$. So there must exist a subset of the two-dimensional number set $\{(i, j),\ 0 \le i \le n-1,\ 0 \le j \le m-1\}$, which makes "$x_i$ is superior to y_j" true if and only if $(i, j) \in H$. If the structure of the subset H is simple, we can construct a certain channel to make "the hacker wins one time" equivalent to "one bit is successfully transmitted from the sender to the receiver." Then, we analyze it using Shannon's "channel coding theorem." For example,

in the game of "rock-paper-scissors," $H = (i, j) : 0 \le i, j \le 2(j - i) \bmod 3 = 2$;

in the game of "coin tossing," $H = (i, j) : 0 \le i = j \le 1$;

in the game of "palm or back," $H = (i, j, k) : 0 \le i \ne j = k \le 1$;

in the game of "finger guessing," $H = (i, j) : 0 \le i, j \le 3(i - j) \bmod 4 = 1$;

in the game of "draw boxing," $H = (x, y, f, g) : 0 \le x, f \le 50 \le g \ne y \le 10x + f = y$.

We have constructed corresponding communication channels for each H above in this paper. However, it is difficult to construct such a communication channel for a general H. But if the above set H can be decomposed into $H = \{(i, j) : i = f(j), 0 \le i \le n - 1, 0 \le j \le m - 1\}$ (namely, the first component j of H is a function of its second component), we can construct a random variable $Z = f(Y)$. Then considering the channel $(X; Z)$, we can give the following equations.

If the "hacker X" attacks with the method xi, and "honker Y" defends with the method y_j in a certain round, then if "X wins," that is, $i = f(j)$, the output of the channel $(X; Z)$ is $Z = f(y_j) = f(j) = i = x_i$. So the output of the channel is the same as its input now; that is, one bit is successfully transmitted from the input of the channel $(X; Z)$ to its output. Conversely, if "one bit is successfully transmitted from the input of the channel $(X; Z)$ to its output," there is "input = output"; that is, "$i = f(j)$", which means "X wins."

Combining the cases above, we obtain the following theorem.

Theorem 16 (the limitation theorem of linear nonblind confrontation). *In the "nonblind confrontation", suppose the hacker X has n attack methods $\{x_0, x_1, \ldots, x_{n-1}\} = \{0, 1, 2, \ldots, n - 1\}$ and the honker Y has m defense methods $\{y_0, y_1, y_{m-1}\} = \{0, 1, 2, \ldots, m - 1\}$, and both sides comply with the rule of winning or losing: "x_i is superior to y_j" if and only if $(i, j) \in H$, where H is a subset of the rectangular set $\{(i, j), 0 \le i \le n - 1, 0 \le j \le m - 1\}$.*

For X, if H is linear and can be written as $H = \{(i, j) : i = f(j), 0 \le i \le n - 1, 0 \le j \le m - 1\}$ (i.e., the first component i of H is a certain function $f(\cdot)$ of its second component j), we can construct a channel $(X; Z)$ with $Z = f(Y)$ to get that, if C is the channel capacity of channel $(X; Z)$, we have the following.

(1) If X wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close to 1 in the k/C rounds.

(2) If X wins S times in n rounds, there must exist $S \le nC$.

For Y, if H is linear and can be written as $H = \{(i, j) : j = g(i), 0 \le i \le n-1, 0 \le j \le m-1\}$ (i.e., the second component j of H is a certain function $g(\cdot)$ of its first component i), we can construct a channel $(Y; G)$ with $G = g(X)$ to get that, if D is the channel capacity of channel $(Y; G)$, we have the following.

(3) If Y wants to win k times, he must have a certain skill (corresponding to the Shannon coding) to achieve the goal by any probability close to 1 in the k/D rounds.

(4) If Y wins S times in n rounds, there must exist $S \le nD$.

6. Conclusion

It seems that these games of nonblind confrontation are different. However, we use an unified method to get the distinctive conclusion; that is, we establish a channel model which can transform "the attacker or the defender wins one time" to "one bit is transmitted successfully in the channel." Thus, "the confrontation between attacker and defender" is transformed to "the calculation of channel capacities" by the Shannon coding theorem [6]. We find that the winning or losing rules sets of these games are linearly separable. For linearly inseparable case, it is still an open problem. These winning or losing strategies can be applied in big data field, which provides a new perspective for the study of the big data privacy protection.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by the National Key Research and Development Program of China (Grant nos. 2016YFB0800602, 2016YFB0800604), the National Natural Science Foundation of China (Grant nos. 61573067, 61472045), the Beijing City Board of Education Science and technology project (Grant no. KM201510015009), and the Beijing City Board of Education Science and Technology Key Project (Grant no. KZ201510015015).

References

[1] R. J. Deibert and R. Rohozinski, "Risking security: policies and paradoxes of cyberspace security," *International Political Sociology*, vol. 4, no. 1, pp. 15–32, 2010.

[2] L. Shi, C. Jia, and S. Lv, "Research on end hopping for active network confrontation," *Journal of China Institute of Communications*, vol. 29, no. 2, p. 106, 2008.

[3] H. Demirkan and D. Delen, "Leveraging the capabilities of service-oriented decision support systems: putting analytics and big data in cloud," *Decision Support Systems*, vol. 55, no. 1, pp. 412–421, 2013.

[4] Y. Yang, H. Peng, L. Li, and X. Niu, "General theory of security and a study case in internet of things," *IEEE Internet of Things Journal*, 2016.

[5] Y. Yang, X. Niu, L. Li, H. Peng, J. Ren, and H. Qi, "General theory of security and a study of hacker's behavior in big data era," *Peer-to-Peer Networking and Applications*, 2016.

[6] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE National Convention Record*, vol. 4, pp. 142–163, 1959.

[7] B. Kerr, M. A. Riley, M. W. Feldman, and B. J. M. Bohannan, "Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors," *Nature*, vol. 418, no. 6894, pp. 171–174, 2002.

[8] K. L. Chung and W. Feller, "On fluctuations in coin-tossing," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 35, pp. 605–608, 1949.

[9] K.-T. Tseng, W.-F. Huang, and C.-H. Wu, "Vision-based finger guessing game in human machine interaction," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '06)*, pp. 619–624, December 2006.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4934082, 9 pages
https://doi.org/10.1155/2017/4934082

Research Article

An Effective Conversation-Based Botnet Detection Method

Ruidong Chen,[1,2] Weina Niu,[1,2] Xiaosong Zhang,[1,2] Zhongliu Zhuo,[1,2] and Fengmao Lv[1]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[2]*Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

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

Received 25 January 2017; Accepted 12 March 2017; Published 9 April 2017

Academic Editor: Lixiang Li

A botnet is one of the most grievous threats to network security since it can evolve into many attacks, such as Denial-of-Service (DoS), spam, and phishing. However, current detection methods are inefficient to identify unknown botnet. The high-speed network environment makes botnet detection more difficult. To solve these problems, we improve the progress of packet processing technologies such as New Application Programming Interface (NAPI) and zero copy and propose an efficient quasi-real-time intrusion detection system. Our work detects botnet using supervised machine learning approach under the high-speed network environment. Our contributions are summarized as follows: (1) Build a detection framework using PF_RING for sniffing and processing network traces to extract flow features dynamically. (2) Use random forest model to extract promising conversation features. (3) Analyze the performance of different classification algorithms. The proposed method is demonstrated by well-known CTU13 dataset and nonmalicious applications. The experimental results show our conversation-based detection approach can identify botnet with higher accuracy and lower false positive rate than flow-based approach.

1. Introduction

Botnet [1] comprises many compromised hosts under the control of the botmaster remotely. Early botnets relied on Internet Relay Chat (IRC) [2] and Hypertext transfer protocol (HTTP) [3] to communicate. The problem of that is single-point invalid and easy to be detected and destroyed. Most botnets are decentralized and use P2P technology [4] to construct command and control (C&C) mechanism [5]. Noncentral node P2P botnet [6] is harder to detect than IRC and HTTP-based one. What is more, bots evolve into attacks which are difficult to track their position. Most current Denial-of-Service (DoS) and spam are caused by botnet [7]. Thus, the botnet is one of the greatest threats to network security.

In the past, researchers used signature-based [8] and anomaly-based intrusion detection systems (IDS) [9, 10] to detect botnet. However, the former has two shortcomings: one is that the original detection rules cannot effectively detect bot program that changes communication means; the other is that inaccurate signatures cause high false positive. Lack of scalability under huge network traffic is another problem.

Currently, the backbone network is based on 1 Gbps or 10 Gbps optical fibers, which renders massive traffic data in short time. Moreover, fast growing P2P applications pose significant strain to data storage. Therefore, identifying botnet traffic under high-speed network is a challenging issue [11]. In this paper, a detection platform with high detection accuracy and powerful traffic processing ability is proposed. It uses conversation-based network traffic analysis and supervised machine learning to identify malicious botnet traffic. The experimental results show that random forest algorithm [12] has higher detection accuracy and lower false positive rate. Moreover, we further explore the top five classifiers (RandomForest, REPTree, RandomTree, BayesNet, and DecisionTump [13]).

The contributions of the paper are threefold. First, a novel botnet detection system with low latency and high accuracy is introduced. Second, our detection method identifies botnet traffic using conversation-based traffic analysis and supervised machine learning. Our approach outperforms the

accuracy based on flow since the false positive rate of botnet traffic decrease is 13.2 percent. In addition to the above two, we evaluate performances of the five well supervised machine learning algorithms (MLAs) [14]. The detection rate of the botnet is up to 93.6%, and the false alarm rate is about 0.3% by the random forest algorithm.

The remainder of this paper is organized as follows: Section 2 gives an overview of botnet detection related works; Section 3 shows the proposed detection method; Section 4 provides the preliminary experimental results; conclusion are summarized in Section 5.

2. Related Work

Botnet detection methods fall into two categories: host behavior-based detection [15] and network-based detection [16].

2.1. Host-Based Detection. Host-based detection is the earliest method. To determine whether a host is compromised, this method continuously monitors the change of process, files, network connections, and registries under a controlled environment [17, 18]. Host-based detection is useful in detecting known bots. However, it performs poorly, because it cannot detect new or variant bots. For example, host-based detection has a sense of inability to identify bots with new technologies like a rootkit, counter debug.

2.2. Network-Based Detection. Network-based detection [19–21] mainly identifies traffic in C&C control phrase of a botnet, because behavior features in this phrase are different from other phrases. Network-based detection mainly focuses on analyzing two kinds of network behaviors: the rate of failed connection and flow features. Most commonly used flow features include the number of uplink (downlink) data packets, the number of uplink (downlink) transmission bytes, the average variance-length of uplink (downlink) data packets, the maximum length of uplink (downlink) data packets, the average variance-length of uplink (downlink) data packets, the duration time of data flow (ms), the rate of the length of data packets in uplink and downlink, and the total length of loaded data packets in a flow. Nowadays, researchers introduce machine learning and neural network to network-based detection to identify unknown botnet traffic. Thus, this method is a hot research point in recognition and analysis of botnet traffic.

Network-based detection method has a high detection rate because it extracts common flow features independent of botnet category. However, in the high-speed and complex network, existing detection platforms based on flow features are ineffective due to high packet drop rate.

3. Our Detection Method

In this section, we describe the components of our proposed botnet traffic detection framework.

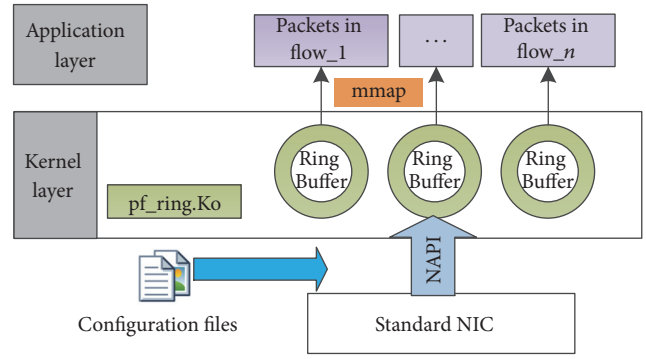


Figure 1: The packet process module architecture.

The framework consists in the following:

(1) Traffic process module for clustering captured packets into different flow buffer

(2) Flow-based feature extraction module for generating statistical characteristics of flow

(3) Conversation-based feature selection module for extracting promising conversation-based feature set

(4) Botnet detection module for identifying botnet traffic using machine learning algorithm.

Packet process module is used to extract the required fields out of the packets. After the extraction of the desired information from the packet process module, the flow-based feature extraction module is used for generating flow features. Based on the flow features, conversation-based feature selection module can obtain promising conversation feature set for the botnet detection module. Botnet traffic detection is accomplished using supervised classification algorithm [22].

3.1. Traffic Process Module. Libcap [23] is used for sniffing the packets from the network interface due to its simple operation and cross-platform. After the NIC captures the packets, Libcap copies packets from the driver to kernel-level using DMA in order to filter them. Then, Libcap copy filtered packets into application in user level for further analyzing, whereas multiple copies of Libcap impose more overhead and consume more time. The mechanics of Libcap makes packet loss and do not reduce the user session. PF_RING [24] is a new network socket that uses New Application Programming Interface (NAPI) and zero copy to capture packet data from a live network. Thus, PF_RING is used to capture traffic onto successive pcaps. The detailed packet capture process is shown in Figure 1.

First, the kernel layer of the packet process module reads the configuration file to set the parameter values, like packet length, ClusterId. ClusterId is the ID of Ring Buffer created by PF_RING. Parameter values are stored in the configuration file so that we can modify them at any time. Second, network devices are turned on and Ring Buffer is created using pfring_open (device, snaplen, flags) function of PF_RING, where device denotes the name of the network device, snaplen denotes the packet length, and flags denotes

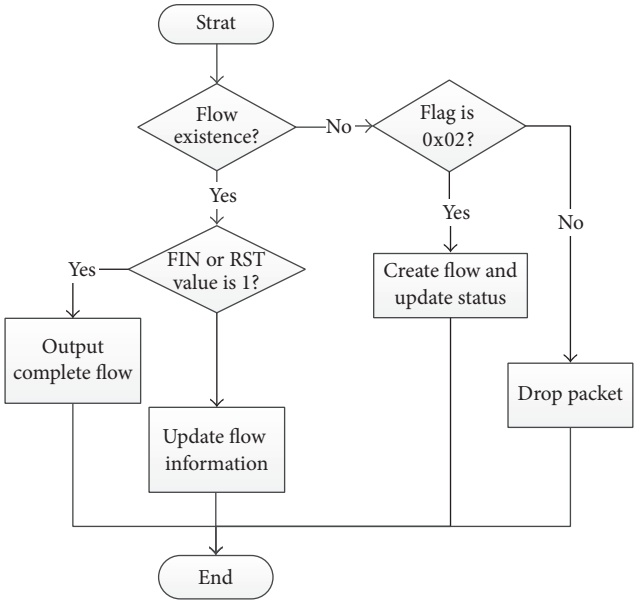


Figure 2: The packet process module architecture.

whether it is in mixed mode. Here, we set snaplen value as 60 because header fields of a packet are needed in this paper. Third, we save the header information, payload length, and arrival time of a packet in different flow buffer according to five tuples (SrcIp, DstIp, SrcPort, DstPort, Proto), which is used to mark a flow. That is, if two different packets have the same source/destination host/port and the same protocol, they belong to the same flow.

3.2. Flow-Based Feature Extraction Module. There are different flow reorganization methods for different transport layer protocols. Using TCP packets as an example, we use a three-way handshake to represent the start of a flow. When a packet whose FIN or RST value is 1 comes, the end of this flow is marked. The detailed TCP flow reorganization process is shown in Figure 2.

When a packet comes, we decide whether the flow this packet belongs to exists. If a packet whose flag value is 0x02, and the flow does not exist, we create a flow according to Ip, protocol, and port. When the flag of the packet takes other values, this packet needs to be dropped. An instance of a flow reorganization state machine can be in only one of the five states: handshake_1, handshake_2, handshake_3, data transmission, and end. If a packet whose flag value is 0x02, this process is in the status of handshake_1. Only when a packet whose flag value is 0x12 is coming, the flow reorganization will be in handshake_2 status. Then, the arrival of a packet whose flag value is 0x10 marks handshake_3 status. After the three-way handshake, data begins transmitting. In the procedure of flow reorganization, whenever there is a packet whose flag value is 0x02, it turns back to the handshake_2 status.

After analyzing the data characters of a botnet, we find that there is a flow similarity of the same botnet. Here, a conversation contains many flows with different source ports. That is, two flows having the same source/destination IP, destination port, and protocol can be classified as the same conversation. Promising conversation feature generating is based on the flow features. Thus, the flow-based feature module extracts statistical features including flow duration, the average interval of up (down) flow, the maximal/minimum/average length of up (down) flow, the number of valid up (down) packets in a flow, the number of transmission bytes of up (down) flows, and the number of small packets in a flow.

3.3. Conversation-Based Feature Selecting Module

3.3.1. Conversation Features

(1) The Duration Time of Flows in a Conversation. The communication between the botmaster and other bot hosts is done by bots. Thus, the duration time of botnet flow is usually fixed and short. However, the duration time of normal flow is determined by user behaviors. Here, we can extract the average duration time of flows in a conversation (avg_duration), the minimum and maximum duration time of flows in a conversation (min_duration, max_duration), the standard deviation of duration time of flows in a conversation (std_duration), and the average arriving intervals of up and down flows in a conversation (avg_finter, avg_binter). Assuming that there are n flows in a conversation, avg_finter = (avg_finter_f_1 + ⋯ + avg_finter_f_n)/n, where avg_finter_f_1 denotes the average arriving intervals of up packets in the first flow.

(2) The Distribution of Flows in Conversation. During the communication process among nodes in a botnet, the size and the number of transmitted data packets are small. And C&C communication flows produced from bot hosts in the same botnet have great similarity [25]. Thus, we extract the average length of up and down flows in a conversation (avg_fpkl, avg_bpkl), the minimum length of up and down flows in a conversation (min_fpkl, min_bpkl), the maximum length of up and down flows in a conversation (max_fpkl, max_bpkl), the standard variation of the length of up and down flows in a conversation (std_avg_fpkl, std_avg_bpkl), the average number of valid up and down flows in a conversation (avg_fpks, avg_bpks), the standard variation of the number of valid up and down flows in a conversation (std_avg_fpks, std_avg_bpks), the average number of transmission bytes of up and down flows in a conversation (avg_fpksl, avg_bpksl), and the standard variation of transmission bytes of up and down flows in a conversation (std_fpksl, std_bpksl).

(3) The Distribution of Small Packets in Conversation. There are many packets within the range of 40320 bytes [26] in the botnet traffic because bots need to constantly connect to new hosts. However, a packet size of the benign server traffic is

large. Thus, the distribution of small packet in a conversation is an interesting characteristic of botnet traffic, like the minimum of small packet in a conversation (min_spacket), the maximum of small packet in a conversation (max_spacket), the average of small packet in a conversation (avg_spacket), and the standard variance of small packet in a conversation (std_spacket). In conclusion, there are 26 features extracted from conversations, which are shown in Table 1.

3.3.2. Feature Selection. We use random forest algorithm [12] to select promising features. All the classification trees in random forest is binary tree. Construction of classification tree meets the principle of recursive splitting from top to bottom. For each classification binary tree, all the train set they used is sampled from the original dataset. In other words, several samples in the original train set may appear many times in the train set of one classification tree and may never appear in any classification tree samples. Algorithm 1 describes how to construct a random forest algorithm in detail.

In the procedure of random forest model establishment, Gini coefficient is used to select feature. Here are 2 classes; thus, the value of K is 2. We suppose that feature A_i ($i = 1, \ldots, 26$) splits dataset D into N parts: $D_1; \ldots; D_n$. On the condition of the feature A_i, Gini coefficient of dataset D is shown as follows:

$$\begin{aligned} \text{Gini}(D, A\_i) &= \sum_{n=1}^{N} \frac{|D_n|}{|D|} \text{Gini}(D_n), \\ \text{Gini}(D_n) &= 1 - \sum_{k=1}^{K} \left(\frac{|C_k|}{|D_n|} \right)^2. \end{aligned} \tag{1}$$

Then, we select promising features according to random forest model. The feature selection process is shown in Algorithm 2. In the algorithm, input data with 27 columns includes 26 conversation features and a class label.

In every iteration, we first rank features according to their importance and then delete the feature with minimum value until detection rate no longer changes. The formula for calculating an RF score of features is shown in (2). In the following equation, there is M decision tree with feature A_i. Gini(D, A_i) indicates Gini coefficient of dataset D using feature A_i in the current decision tree.

$$\text{RF\_score}(A\_i) = \sum_{m=1}^{M} K * \prod_{n=1}^{N_{-m}} \frac{|D_n|}{|D|} - \text{Gini}(D, A\_i). \tag{2}$$

Depending on the following random forest model, the detection rate is generated using testing data. In this work, we use the features including {std_bpksl, std_avg_fpkl, std_avg_fpks, std_f(b)pksl, std spacket}. Feature vectors constructed in this paper include {SrcIp, DstIp, DstPort, Pro, std_bpksl, std_avg_fpkl, std_avg_fpks, std_f(b)pksl, std_spacket}, and using {SrcIp, DstIp, DstPort, Proto} represents the conversion of visiting the same service.

Table 1: Conversation features.

Feature value	Description of feature value
avg_duration	The average duration time of flows in a conversation
min_duration	The minimum duration time of flows in a conversation
max_duration	The maximum duration time of flows in a conversation
std_duration	The standard deviation of duration time of flows in a conversation
avg_f(b)inter	The average interval of up (down) flows in a conversation
avg_f(b)pkl	The average length of up and down flows in a conversation
min_f(b)pkl	The minimum length of up (down) flows in a conversation
max_f(b)pkl	The maximum length of up (down) flows in a conversation
std_avg_f(b)pkl	The standard variation of the length of up (down) flows in a conversation
avg_f(b)pks	The average number of up (down) valid flows in a conversation
std_avg_f(b)pks	The standard variation of the number of up (down) valid flows in a conversation
avg_f(b)pksl	The average of transmission bytes of up (down) flows in a conversation
std_f(b)pksl	The standard variation of transmission bytes of up (down) flows in a conversation
min_spacket	The minimum of small packet in a conversation
max_spacket	The maximum of small packet in a conversation
avg_spacket	The average of small packet in a conversation
std_spacket	The standard variance of small packet in a conversation

3.4. Botnet Detection Module. In order to achieve scalability in botnet detection module, we use API provided by Weka to implement machine learning algorithms [14]. The conversation feature need be saved in CSV format at the conversation-based feature selecting

Input: *data* a labeled dataset with p features
Output: PF a random forest model
(1) $n \leftarrow$ the number of decision trees, $m \leftarrow$ the number of selected features
(2) initialization $m = M$, $n = N$, $i = 1$
(3) while $i \leq N$ do
(4) draw a bootstrap sample Z^* of size S from data
(5) repeat
(6) select M features at random from the p features
(7) calculate the Gini coefficient of selected M features
(8) select the feature with lower Gini coefficient among the M
(9) split the node into two daughter nodes
(10) **until** the minimum node size is reached
(11) construct decision tree i
(12) $i = i + 1$
(13) **end while**

Algorithm 1: Random forest algorithm.

Input: *train_data* a labeled training set, *test_data* a labeled testing set
Output: PF a list of promising features
(1) $\delta \leftarrow$ an error range, *BD_l* $\leftarrow$ the botnet traffic detection rate, *BD_c* $\leftarrow$ the current botnet traffic detection rate
(2) initialization $i = 1$, $\delta = \Theta$
(3) *BD_l* = *BD_c* = Randonforest (all features)
(4) **while** $|BD\_c - BD\_l| \leq \Theta$ **do**
(5) *BD_l* = *BD_c*
(6) calculate RF scores of importance
(7) rank the RF scores
(8) delete the feature with the smallest importance from *train_data* and *test_data*
(9) *BD_c* = randomforest (remaining_features)
(10) **end while**

Algorithm 2: Feature selection algorithm.

module. First, the module reads feature vectors using weka.core.converters.ConverterUtils.DataSource. However, some features like SrcIp, DstIp, DstPort, and Proto have no efficiency in identifying botnet traffic. Second, this module deletes them using weka.core.Instances. Third, we use random forest algorithm to train these data through weka.classifiers.trees.RandomForest. Fourth, this module uses the trained classifier to predict unlabeled data by calling classifyInstance(unlabeled.instance (i)) function. Here, "unlabeled" denotes testing data without a label.

4. Experimental Results and Performance Analysis

4.1. Experimental Setup. Famous public datasets used to detect botnet traffic include dataset disclosed from Information Security and Object Technology (ISOT) organization [25], Stratosphere [27], and the CTU University [28]. The dataset from Stratosphere contains many types of botnet behaviors traffic, such as the traffic of scanned port, traffic of C&C communication, and attack traffic. However, most botnet traffic of this dataset is IRC and HTTP botnet, and there is only one type of P2P botnet traffic. The dataset from ISOT only contains three types of botnet traffic, Waledac, Storm, and Zeus, and many background traffic. The dataset from the CTU University consists of thirteen scenarios of different botnet samples. Thus, in the experiment, we use dataset from CTU University. And the distributions of botnet types about training and test in our experiment are listed in Tables 2 and 3. For example, Rbot contains three types of botnets, namely, IRC, DDoS, and the US.

4.2. The Results and Analysis of Experiments. During the process of experiment, we assess our detection method by adopting the train set and test set from CTU13. The CUT13 dataset provides a better test environment for unknown botnet because this test set contains many types of botnet traffic which do not exist in the training set.

The effectiveness of the top five classifiers, namely, random forest, REPTree, randomTree, BayesNet, and DecisionTump [29], has been studied with the CTU botnet traffic and

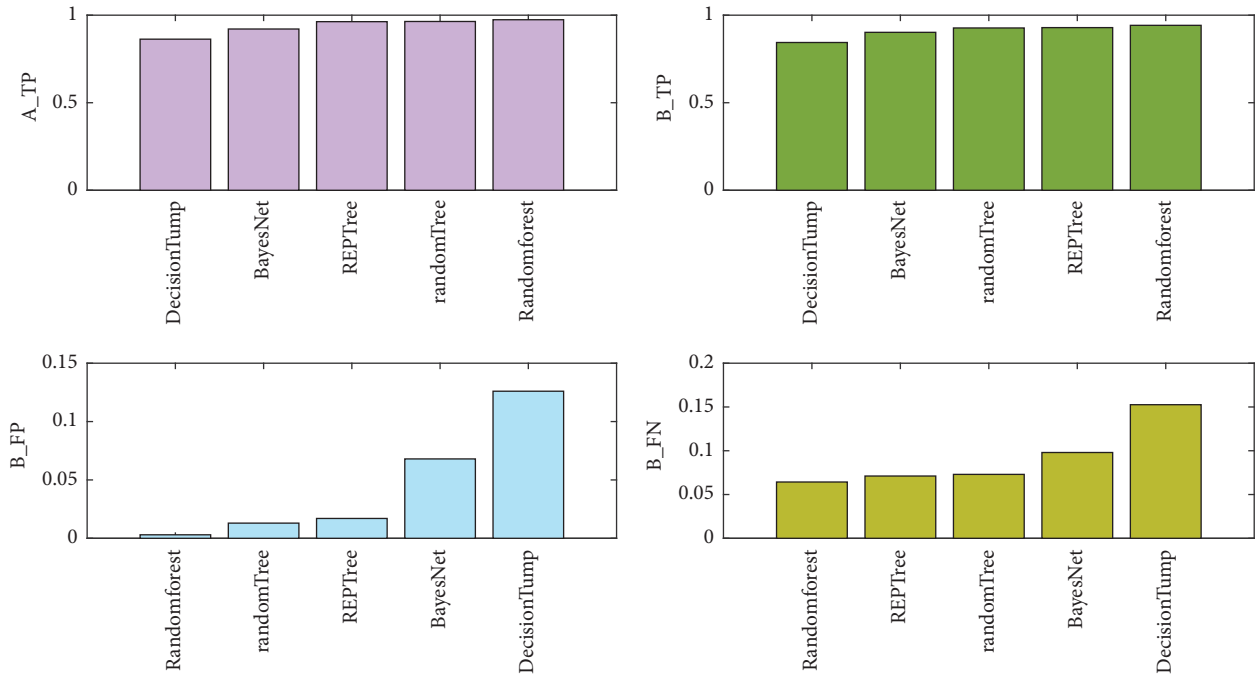


Figure 3: Detection rate of the top five classifiers.

Table 2: Distribution of botnet types in the training dataset.

Botnet name	Type	Portion of dataset
Rbot	IRC, DDoS, US	0.1%
Virut	SPAM, PS, HTTP	0.485%
Menti	PS	3.89%
Sogou	HTTP	0.035%
Murlo	PS	1.64%
Neris	IRC, SPAM, CF, PS	31.3%

Table 3: Distribution of botnet types in the testing dataset.

Botnet name	Type	Portion of dataset
Neris	IRC, SPAM, CF	3.21%
Rbot	IRC, PS, US	2.646%
Rbot	IRC, DDoS, US	0.088%
Virut	SPAM, PS, HTTP	0.4%
Menti	PS	3.33%
Sogou	HTTP	0.036%
Murlo	PS	1.4%
Neris	IRC, SPAM, CF, PS	28.9%
NSIS.ay	P2P	1.71%
Virut	SPAM, PS, HTTP	1.07%

normal traffic generated by benign programs. The detailed contrast tests are done in WEKA, in terms of A_TP, B_TP, B_FP, and B_TN, explained as follows: the ratio of benign traffic and botnet traffic recognized correctly, the ratio of botnet traffic detected as botnet conversation, the ratio of begin traffic classified as botnet traffic, and the ratio of botnet traffic identified as normal traffic. They are defined as follows:

$$\begin{aligned} \text{A\_TP} &= \frac{\text{TP} + \text{TN}}{\text{TM} + \text{TB}}; \\ \text{B\_TP} &= \frac{\text{TP}}{\text{TM}}; \\ \text{B\_FP} &= \frac{\text{TN}}{\text{TP}}; \\ \text{B\_TN} &= \frac{\text{FN}}{\text{TM}}, \end{aligned} \tag{3}$$

where true positive (TP) indicates that the number of botnet conversations is correctly classified; true negative (TN) indicates that the number of benign traffic conversations is correctly classified; false positive (FP) expresses that the number of benign traffics is detected as botnet traffic; false negative (FN) indicates that the number of botnet traffics is detected as benign traffic; TM indicates the total number of botnets, and TB expresses the total number of benign traffics.

The experiment result is shown in Figure 3.

The whole recognition rate of DecisionTump is the lowest because there is a one-level decision tree in the DecisionTump. Random forest algorithm selects variables automatically during the model formation and establishes the optimal discriminant model. Thus, the detection rate of random forest algorithm is the highest. Meanwhile, random forest has a lower false positive and false negative rates than the other four. Moreover, there is no obvious difference among the

Table 4: Experimental parameters settings.

Transmit speed (Gbps)	Internal time (s)			
	30		60	
	The number of flows	The number of conversations	The number of flows	The number of conversations
1	138825	39734	203741	61380
10	261630	92933	452930	158722

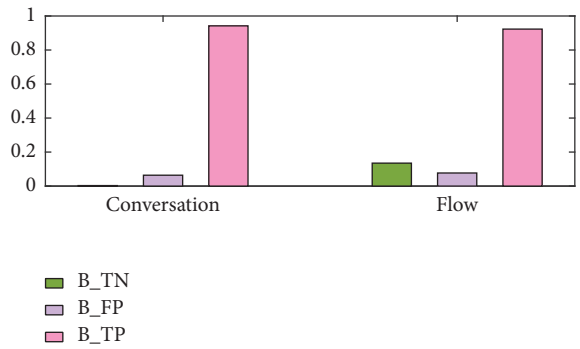


Figure 4: Detection effect of flow-based and conversation-based features.

detection effect of BayesNet, REPTree, and randomTree. The botnet traffic detection of Decision-Tump is 84.4%. However, the detection accuracy of the other four algorithms is more than 90%. The false positive rate and true negative rate of the top five algorithm are under 10% except for DecisionTump.

Kirubavathi and Anitha [20] proposed a botnet detection method via mining of traffic flow characteristics. In their work, they used features like small packets, packet ratio, initial packet length, and bot response packet to identify botnet traffic. Here, we compare the detection rates of flow feature and conversation feature. The result is shown in Figure 4.

As it can be seen from Figure 4, the false positives rate of conversation-based detection and flow-based detection is 0.3% and 13.5%, respectively. Thus, the experimental results show that the false positives rate of our proposed method decreases more than ten times. Meanwhile, the botnet identification rate of our method does not reduce.

In theory, the higher the number of classification trees, the higher the classification accuracy rate. However, if the number and depth of classification tree are extremely high, they will reversely affect the classification speed of classifier. In order to determine the two parameter values of the number and depth of classification tree from random forest algorithm in this paper, we analyze the influence on the classification accuracy by adjusting parameters. In the experiment, the number of classification trees can be set as 10, 50, 100, and 200, and the depth of each classification tree can be set as 2, 4, 10, 20, and so forth. The experiment results of different classification tree size and different classification tree depth are shown in Figure 5.

When the number of the classification trees is 100 and the depth is 10, the detection rate of random forest algorithm reaches the maximum. Afterward, regardless of increasing the number or the depth of the classification trees, the detection rate does not increase anymore. Thus, when the number of the classification trees is set as 100, and the depth of classification tree is set as 10 in the experiment, the random forest works the best.

4.3. Online P2P Botnet Traffic Detection Platform. Our framework has been implemented in Python and utilizes Microsoft Network Monitor to capture packets from a network interface or a pcap file. Because the timeout value of TCP/UDP packets is 60 s, we set the time window as 60 s in this paper to extract conversation feature. While we experimented with different time window settings, the 60-second time window showed the best accuracy at considerably low computational complexity. In the high-speed network environment, we count the number of conversations and the data flows contained in the interval of 60 s and gather the 1 Gbps and 10 Gbps network in many times. The interval of the gathering is 60 s and 30 s, and then we compute the average value. The result is shown in Table 4. In Table 4, T stands for time, S stands for speed, and conversa stands for conversation.

According to Table 4, in the 10 Gbps network, and the interval of 60 s, the average number of passing flows is 452930 and the average number of conversations is only 158722. Thus, using the conversation features can greatly reduce the number of feature vectors. The reason of that is a conversation consists of any number of flows that have the same source/destination host/port and the same protocol. According to the foregoing experiments, we can see that the time of using random forest algorithm to detect 204711 feature vectors is 27.1 seconds. Thus, half real-time botnet detection platform based on random forest classifier and conversation features can identify botnet traffic under the high-speed network environment.

5. Conclusion and Future Work

In this paper, we propose an efficient botnet traffic detection system which can handle heavy network bandwidths. Our framework utilizes PF_RING to solve the high packet drop rate of Libcap. RF-RING has low latency and low overhead to extract required fields of traffic. Then, feature selection is conducted to reduce the dimensionality of data. Conversation features combine the advantages of the existing detection methods based on flow statistical behaviors and flow similarity. We select promising features using random forest algorithm in order to reduce the feature dimension. This

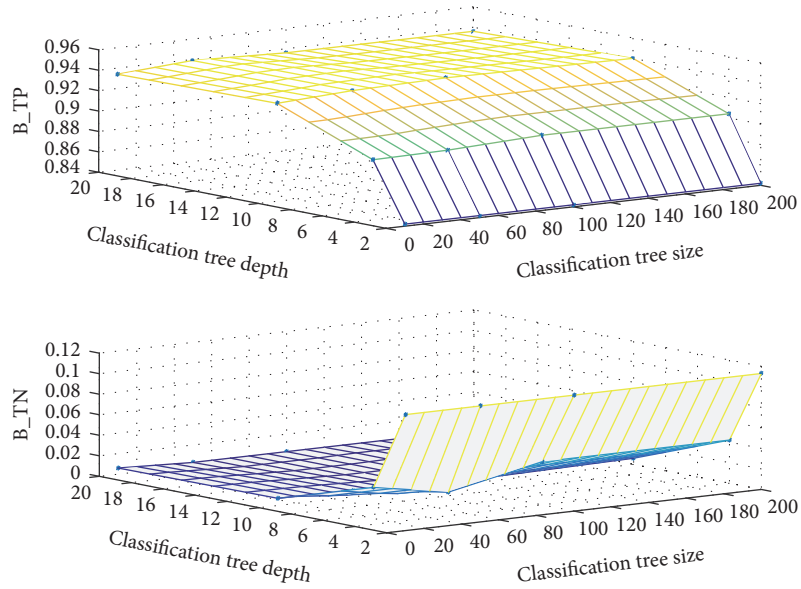


Figure 5: Detection rate for different number and different depth of classification trees.

framework selects the machine learning which obtained the best learning performance. The experiments are conducted on the offline public dataset and online real data. The experimental results show that conversation features used in this paper behave better than flow features in the CTU13 open source dataset. Among all the classification algorithms, the detection rate of random forest is the highest, which is up to 93.6%. And the false alarm rate is only 0.3%, which is ten times less than detection based on traffic flow characteristics.

The future work will focus on mining association rules according to our proposed conversation features. Moreover, we need to further identify specific botnet categories in order to design corresponding defense plans.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61572115, 61502086, and 61402080) and the Key Basic Research of Sichuan Province (Grant no. 2016JY0007).

References

[1] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han, "Botnet research survey," in *Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC '08)*, pp. 967–972, IEEE, August 2008.

[2] C. Mazzariello, "IRC traffic analysis for botnet detection," in *Proceedings of the 4th International Conference on Information Assurance and Security (IAS '08)*, pp. 318–323, IEEE, September 2008.

[3] J.-S. Lee, H. C. Jeong, J.-H. Park, M. Kim, and B.-N. Noh, "The activity analysis of malicious http-based botnets using degree of periodic repeatability," in *Proceedings of the International Conference on Security Technology (SECTECH '08)*, pp. 83–86, IEEE, December 2008.

[4] W. Zhou and X. Wu, "Survey of p2p technologies," *Computer Engineering and Design*, vol. 27, no. 1, pp. 76–79, 2006.

[5] H. R. Zeidanloo and A. A. Manaf, "Botnet command and control mechanisms," in *Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE '09)*, pp. 564–568, IEEE, December 2009.

[6] D. Dittrich and S. Dietrich, "P2P as botnet command and control: a deeper insight," in *Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE '08)*, pp. 41–48, IEEE, October 2008.

[7] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '09)*, pp. 268–273, IEEE, June 2009.

[8] R. Villamarín-Salomón and J. C. Brustoloni, "Bayesian bot detection based on DNS traffic similarity," in *Proceedings of the 24th Annual ACM Symposium on Applied Computing (SAC '09)*, pp. 2035–2041, ACM, March 2009.

[9] S. Arshad, M. Abbaspour, M. Kharrazi, and H. Sanatkar, "An anomaly-based botnet detection approach for identifying stealthy botnets," in *Proceedings of the IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE '11)*, pp. 564–569, IEEE, December 2011.

[10] M. N. Sakib and C.-T. Huang, "Using anomaly detection based techniques to detect HTTP-based botnet C&C traffic," in *Proceedings of the IEEE International Conference on Communications (ICC '16)*, pp. 1–6, IEEE, Kuala Lumpur, Malaysia, May 2016.

[11] P. V. Amoli and T. Hämäläinen, "A real time unsupervised NIDS for detecting unknown and encrypted network attacks in high

speed network," in *Proceedings of the 2nd IEEE International Workshop on Measurements and Networking (M & N '13)*, pp. 149–154, IEEE, October 2013.

[12] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," *Information Sciences*, vol. 278, pp. 488–497, 2014.

[13] S. Kalmegh, "Analysis of WEKA data mining algorithm REPTree, simple CART and RandomTree for classification of Indian news," *International Journal of Innovative Science, Engineering, and Technology*, vol. 2, no. 2, pp. 438–446, 2015.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[15] S. Saad, I. Traore, A. Ghorbani et al., "Detecting P2P botnets through network behavior analysis and machine learning," in *Proceedings of the 9th Annual International Conference on Privacy, Security and Trust (PST '11)*, pp. 174–180, IEEE, Montreal, Canada, July 2011.

[16] M. R. Rostami, B. Shanmugam, and N. B. Idris, "Analysis and detection of P2P botnet connections based on node behaviour," in *Proceedings of the World Congress on Information and Communication Technologies (WICT '11)*, pp. 928–933, IEEE, December 2011.

[17] H. Zhang, M. Gharaibeh, S. Thanasoulas, and C. Papadopoulos, "Botdigger: detecting DGA bots in a single network," in *Proceedings of the IEEE International Workshop on Traffic Monitoring and Analaysis*, Louvain La Neuve, Belgium, April 2016.

[18] W. Wang, B.-X. Fang, and X. Cui, "Botnet detecting method based on group-signature filter," *Journal on Communications*, vol. 31, no. 2, pp. 29–35, 2010.

[19] K. Shanthi and D. Seenivasan, "Detection of botnet by analyzing network traffic flow characteristics using open source tools," in *Proceedings of the 9th IEEE International Conference on Intelligent Systems and Control (ISCO '15)*, pp. 1–5, IEEE, January 2015.

[20] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics," *Computers and Electrical Engineering*, vol. 50, pp. 91–101, 2016.

[21] J. Zhang, R. Perdisci, W. Lee, X. Luo, and U. Sarfraz, "Building a scalable system for stealthy P2P-botnet detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 27–38, 2014.

[22] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '14)*, pp. 797–801, IEEE, February 2014.

[23] L. M. Garcia, "Programming with libpcap—sniffing the network from our own application," *Hakin9-Computer Security Magazine*, p. 2-2008, 2008.

[24] M. M. Rathore, A. Ahmad, and A. Paul, "Real time intrusion detection system for ultra-high-speed big data environments," *The Journal of Supercomputing*, vol. 72, no. 9, pp. 3489–3510, 2016.

[25] D. Zhao, I. Traore, B. Sayed et al., "Botnet detection based on traffic behavior analysis and flow intervals," *Computers and Security*, vol. 39, pp. 2–16, 2013.

[26] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by capturing group activities in network traffic," in *Proceedings of the 4th International ICST Conference on Communication System Software and Middleware*, p. 2, ACM, June 2009.

[27] P. Judge, D. Alperovitch, and W. Yang, "Understanding and reversing the profit model of spam (position paper)," in *Proceedings of the 4th Workshop on the Economics of Information Security*, June 2005.

[28] F. Haddadi, D.-T. Phan, and A. N. Zincir-Heywood, "How to choose from different botnet detection systems?" in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS '16)*, pp. 1079–1084, IEEE, April 2016.

[29] A. Sharma and S. K. Sahay, "An effective approach for classification of advanced malware with high accuracy," *International Journal of Security and Its Applications*, vol. 10, no. 4, pp. 249–266, 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4916953, 9 pages
https://doi.org/10.1155/2017/4916953

Research Article

Identifying APT Malware Domain Based on Mobile DNS Logging

Weina Niu,[1,2] Xiaosong Zhang,[1,2] GuoWu Yang,[2] Jianan Zhu,[3] and Zhongwei Ren[1]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[2]*Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[3]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China*

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

Received 25 January 2017; Accepted 7 March 2017; Published 6 April 2017

Academic Editor: Lixiang Li

Advanced Persistent Threat (APT) is a serious threat against sensitive information. Current detection approaches are time-consuming since they detect APT attack by in-depth analysis of massive amounts of data after data breaches. Specifically, APT attackers make use of DNS to locate their command and control (C&C) servers and victims' machines. In this paper, we propose an efficient approach to detect APT malware C&C domain with high accuracy by analyzing DNS logs. We first extract 15 features from DNS logs of mobile devices. According to Alexa ranking and the VirusTotal's judgement result, we give each domain a score. Then, we select the most normal domains by the score metric. Finally, we utilize our anomaly detection algorithm, called Global Abnormal Forest (GAF), to identify malware C&C domains. We conduct a performance analysis to demonstrate that our approach is more efficient than other existing works in terms of calculation efficiency and recognition accuracy. Compared with Local Outlier Factor (LOF), k-Nearest Neighbor (KNN), and Isolation Forest (iForest), our approach obtains more than 99% F-M and R for the detection of C&C domains. Our approach not only can reduce data volume that needs to be recorded and analyzed but also can be applicable to unsupervised learning.

1. Introduction

Advanced Persistent Threat (APT) [1, 2] is an attack that is launched by the well-funded and skilled organization to steal high-value information for a long time. APT attackers would install malware on the compromised machine to build command and control (C&C) channel after infiltrating into the targeted network. Most malware makes use of Domain Name System (DNS) to locate their domain name servers and compromised devices. Then, APT attackers can establish long-term connection to victims' devices for stealing sensitive data. Thus, malware C&C domain detection can help security analysts to block essential stage of APT.

Currently, there are some works to identify C&C domain by analyzing network traffic about PC [3–8]. BotSniffer [3], BotGAD [4], and BotMiner [5] made use of specific behavior anomaly (e.g., daily similarity and short life) to detect C&C involved in a botnet. The main reason is that bot hosts have group similarity. Other works [6–8] also distinguish between malicious domains and normal domains according to domain-based features, such as domain name string composition, registration time, and active time. However, these detection approaches cannot be applied to APT malware since APT attackers infect a small number of machines, and they behave normally to avoid detection. Machine learning technology is proved to be effective in identifying malware [6]. However, there are few artificially marked data of APT malware. Moreover, normal and abnormal samples overlap with each other.

In order to address these challenges, we propose an approach to identifying APT malware domains based on DNS logs. We conduct experiments to evaluate our proposed algorithm, called Global Abnormal Forest (GAF), with three traditional algorithms, namely, Local Outlier Factor (LOF),

k-Nearest Neighbor combined with LOF (LOF-KNN), and Isolation Forest (iForest). The experimental results demonstrate that our proposed algorithm behaves best on a dataset consisting of 300000 DNS requests each day from a regional base station. Specifically, the contributions of this work are specified as follows:

(i) We characterize statistics of normal domains and define a rule based on Alexa and VirusTotal to select the most normal domains.

(ii) We extract 15 features of mobile DNS requests in multigranularity by studying large DNS logs in a real dynamic network environment consisting of 10K devices with more than 300,000 DNS requests per day.

(iii) We propose an anomaly detection algorithm to compromise accuracy and efficiency of C&C domains detection by introducing differentiated information entropy.

The structure of this paper is arranged as follows. we motivate the need for APT malware C&C detection using anomaly detection in Section 2; Section 3 presents an overview of the proposed approach and introduces the most normal domain identification rules, and we motivate the choice for features that are related to APT malware C&C domain in Section 3; Section 4 describes the building of our anomaly detection model; Section 5 completes experimental evaluation metrics and illustrates the experimental results of different algorithms; Section 6 introduces the related work; Section 7 makes a conclusion of the paper.

2. Background on C&C Detection Using Anomaly Detection

APT was first used in 2006 and has become widely known since the exposure of Google Aurora in 2010 [7]. In 2013, the APT attack was pushed to cusp due to PRISM. Thus, the APT attack has brought new challenges to cybersecurity due to long-latent, intelligence penetration and overcustomization [8, 9]. APT attackers often install DNS-based APT malware, for instance, Trojan horse or backdoor, on the infected machine for stealing sensitive data and hiding the real attack source. Identifying malware during their command control channel establishment phase is a good choice. However, DNS behavioral features of compromised machines infected by APT malware are different from the botnet. Thus, APT malware identification based on DNS data is a challenge.

Suspicious instances of APT malware are rare and the amount of data cannot be fully labeled by the expert. The most normal domain instances within the DNS data are available. Moreover, anomaly detection [10] can identify new and unknown attack since it does not depend on fixed signatures. Thus, we use anomaly detection to identify malware C&C domain using mobile DNS logs. The most common anomaly detection includes statistical anomaly detection, classification-based anomaly detection, and clustering-based anomaly detection [11]. If the labeled set has been collected, classification-based anomaly detection, like Genetic Algorithm [12], Support Vector Machine [13], and Neural Network [14], is preferable. However, in the real APT attack, the label of data is very difficult to obtain. The unsupervised method can be used to identify malware C&C domain, such as LOF, LOF-KNN, and iForest. LOF [15] determines whether the data is an outlier according to neighbor density. LOF-KNN [16] identifies outlier according to similarity. However, these two approaches have high computational complexity and too many false alarms. To ease these two problems, iForest [17] detects anomalies using the average path length of trees that requires a small subsampling size to achieve high detection performance. Thus, we can build partial models and exploit subsampling to identify malware C&C domain. Isolation Forest is based on the assumption that each instance is isolated to an external node when a tree is grown. Unfortunately, attribute values of normal domain and malware domain are relatively close. Moreover, traditional anomaly detection algorithms ignore the different influences of different properties. In this work, we introduce differentiated information entropy to improve the efficiency and utilize distance measures to detect anomalies.

3. Overview of Our Approach

In this section, we present an overview of the proposed approach for identifying APT malware domain, explain why we select those features that may be indicative of APT malware domain, and illustrate the metric for selecting the most normal domains.

3.1. Architecture of Our Approach. DNS logs are small but important. Thus, this work mainly focuses on the analysis of DNS logs in order to detect suspicious domains involved in APT malware. We store DNS logs that contain accessing user, source IP, destination IP, country flag, domain name, request time, and response time. Then we extract features according to logs and make use of anomaly detection technology to identify APT malware C&C domain. Figure 1 gives an overview of the system architecture of the proposed approach. The system consists of components including the following: (1) DNS logs collector stores the DNS logs produced by mobile devices in the network that is being monitored; (2) multigranularity feature extractor is responsible for extracting features of domains that are stored in DNS log database; (3) normal domain identifier is used to select the most normal domains; (4) anomaly learning module trains anomaly detector using malware domain that is labeled by experts from grey set and APT malware C&C domain produced by detector, normal instance from normal set; (5) anomaly detector takes decisions according to the identification results produced by the anomaly detection model.

The deployment of the system consists of three steps. In the first step, the features that we interested are extracted. Details and motivations on the chosen features will be discussed in Section 3.2. The second step defines a metric to select normal domain used to train. The third step involves

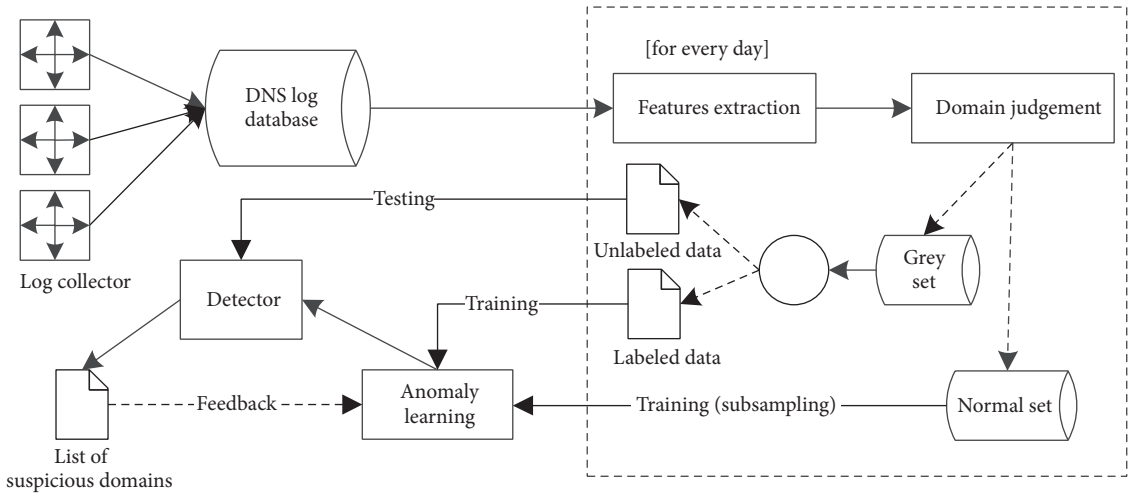


Figure 1: Framework of our proposed identification approach.

Table 1: Features of domain name.

FeatureSet	FeatureName
DNS request and answer-based features	Number of distinct source IP addresses
	Number of distinct IP addresses with the same domain
	IP in the same country
	using the predefined IP addresses
Domain-based features	Alexa ranking
	The length of domain
	The level of domain
	containing IP address
Time-based features	Request frequency
	Reaction time
	repeating pattern
whois-based features	Registration duration
	Active duration
	Update duration
	Number of DNS

our proposed anomaly detection algorithm, which uses part of normal samples to predict C&C domains. The proposed algorithm is described in detail in Section 4. The result is a list of the suspicious domains involved in APT malware.

3.2. Feature Extraction. In this work, we extracted 15 features to detect APT malware C&C domains based on mobile DNS logs. We also gave explanations of the 15 features and explained the reasons that they can be used to detect malicious domain. The extracted domain features are shown in Table 1.

3.2.1. DNS Request and Answer-Based Features. APT attackers usually use servers residing in different countries to build C&C channel in order to evade detection. Moreover, attackers make use of fast flux to hide the true attack source [18]. APT attacker changes the C&C domain to point to predefined IP addresses, such as look back address and invalid IP address. With this insight, we extracted three features from DNS request and response, such as the number of distinct source IP addresses, the number of distinct IP addresses with the same domain, IP in the same country, and using the predefined IP addresses.

3.2.2. Domain-Based Features. Attackers prefer to use the long domain to hide the doubtful part [19]. By analyzing the network traffic produced during the malware communicates with command and control servers, we find that many malware C&C domains have the following characteristics: high level, long string, containing IP address, and low visitor number. Thus, Alexa ranking, the length of the domain, the level of domain, and containing IP address are helpful in identifying malware domain. For example, if a domain name contains an IP address, such as "192.168.1.173.baidu.com", we would conclude that it may be a malicious domain.

3.2.3. Time-Based Features. When there is a connecting failure in the process of compromised device connect to the C&C server, compromised machine may send many repeated DNS requests. Sometimes, behaviors of these infected devices show similarities. Since IP address of malware domain is not stored in the local server, the domain name resolution takes longer time. Moreover, we observe that few domains have high query frequency through analyzing the domain access records during one day in our experimental environment, which is illustrated in Figure 2. This phenomenon helps us to further identify malicious domain names. Thus, we extracted three features to identify APT malware C&C domain, such as request frequency, reaction time, and repeating pattern.

3.2.4. Whois-Based Features. Trustworthy domains are regularly paid for several years in advance and they have a long

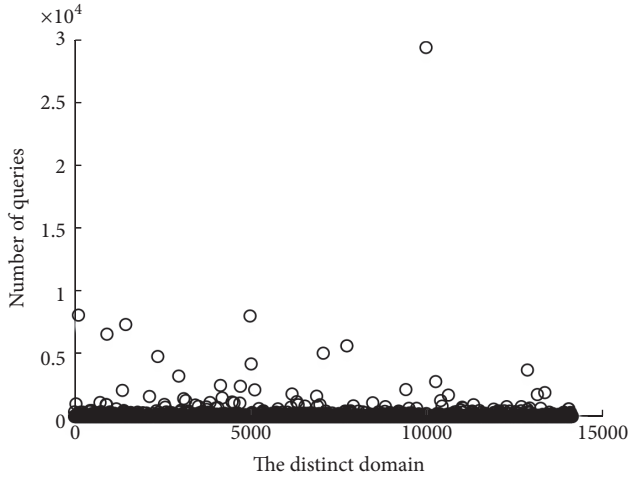


Figure 2: Distribution of query frequency of distinct domain.

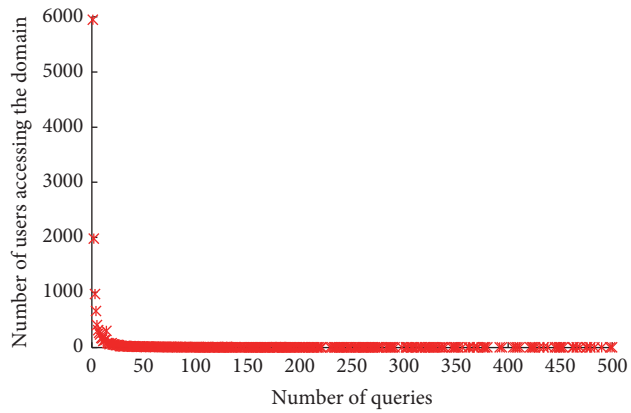


Figure 3: Distribution of the number of domain queries initiated by internal devices.

time to live [20]. However, most malware domains live for a short period of time, which is less than 6 months. Moreover, DNS record of the suspicious domain is empty or not found. Based on the above observation, we can use registration duration, active duration, update duration, and DNS record to detect malicious domain.

3.3. A Metric for Normal Domain Judgement. In order to implement anomaly detection, it is necessary to determine normal samples. An intuitive approach for selecting normal domains according to the number of DNS requests initiated by internal devices. However, in order to reduce exposure risk, APT attackers do not make use of malware C&C server to control too many infected machines. Moreover, in our experimental environment consisting of about 10K mobile devices, the distribution of the number of domains queried by internal devices during one day follows heavy-tailed distributions, as shown in Figure 3. There are about half-domains were queried each time. Thus, we can conclude that the number of distinct access devices cannot effectively identify the normal domain. By analyzing APT malware, we find that malicious domain ranked above the top 200,000 [21]. Thus, the number of visitors and the number of pages they visit are a feature used to identify the normal domain. Furthermore, VirusTotal aggregates numerous antivirus products and online scan engine to check for the malicious domain. Thus, we use Alexa ranking and VirusTotal results to judge normal domains, whose Alexa ranking is below 200,000 in international domains and 30,000 in domestic domains, and VirusTotal's test result is less than 3.

4. Building Anomaly Detection

In this section, we explained our anomaly detection algorithm, called GAF.

Definition 1 (global abnormal tree). Let T be the center of a global abnormal tree. N is the number of samples in this global abnormal tree. A test, which consists of d-variate such that the test has a larger distance from T, is an outlier.

Given a dataset $X = (x_1, x_2, \ldots, x_m)$ of m normal samples with d-dimension features, in other words, $x_i = (f_i^1, f_i^2, \ldots, f_i^d)$, the global abnormal tree building process is illustrated as follows. Firstly, we select N normal samples without replacement from the dataset X to build training set $X' = (x_1, x_2, \ldots, x_n)$. Secondly, we calculate the weight of each feature through introducing differentiated information entropy. Thirdly, we select the center of the N normal samples according to

$$T = \left(\sum_{i=1}^{n} \frac{f_i^1}{n}, \sum_{i=1}^{n} \frac{f_i^2}{n}, \ldots, \sum_{i=1}^{n} \frac{f_i^d}{n} \right). \tag{1}$$

An abnormal domain is acquired according to the distance from the node p to the center of the global abnormal tree, which can be calculated using (2). As it is illustrated in (3), once the mean distance of tester is larger than the threshold value T_r, it can be denoted as a suspicious domain.

$$d(p, T) = \sqrt{\sum_{i=1}^{d} \omega_i \left(f_p^i - f_T^i \right)^2 f^i} \tag{2}$$

$$M_d = \frac{\sum_{i=1}^{N} d(p, T_i)}{N} > T_r. \tag{3}$$

In order to identify the weight of each feature, we need to calculate information entropy of each feature using (4), where k represents k distinct values of normal samples in the i_{th} dimension and x_j^i represents the number of normal samples in the i_{th} dimension whose value equals the j_{th} value. Then, each feature splits set into two parts: $\{f^i\}$ and $\{S - f^i\}$. Thus, the information entropy difference is calculated by (5), which

Input: N: The number of Global Abnormal Tree, M: The number of normal sub-samples used in each Global Abnormal Tree, $X = (x_1, x_2, \ldots, x_n)$: The normal samples, $Y = (y_1, y_2, \ldots, y_k)$: The gery samples
Output: L: The list of suspicious domains
(1) **For** Global Abnormal Tree T_i $(i = 1, 2, \ldots, N)$
(2) Select M sub-samples from X without replacement: $X_i = (x_1, x_2, \ldots, x_M)$
(3) Calculate information entropy of each feature $E(f^i)$ $(i = 1, 2, \ldots, d)$
(4) **For** each feature f^i $(i = 1, 2, \ldots, d)$
(4.1) Calculate information entropy difference of each feature $\Delta E(f^i)$ $(i = 1, 2, \ldots, d)$
(4.2) Set feature weight $\omega_i = \Delta E(f^i)$
(4.3) Compute standard feature weight ω_i
(5) Calculate the center of T_i using normalization sub-samples
(6) Calculate the distance from sample y_i $(i = 1, 2, \ldots, k)$ in Y from the center of T_i
(7) **End for**
(8) Calculate the mean distance M_d
(9) Identify abnormal according to $M_d > T_r$

Algorithm 1: GAF.

is used to represent feature weight. In (5), the feature weight is normalized.

$$E\left(f^i\right) = \sum_{j=1}^{k} \frac{x_j^i}{n} \log \frac{x_j^i}{n} \tag{4}$$

$$\Delta E\left(f^i\right) = \frac{\sum_{j=1}^{d} E\left(f^j\right)}{n} - \left(E\left(f^i\right) + \frac{\sum_{j=1, j\neq i}^{d} E\left(f^j\right)}{n-1} \right). \tag{5}$$

In the process of anomaly detection based on global outlier factor, the tester is classified as abnormal according to the distance to the center of distinct global abnormal tree. In each tree, the centroid is calculated according to the normal samples selected from training test. And the weight of each feature in the different tree is calculated according to the current normal instances. The pseudocode of GAF algorithm is shown in Algorithm 1.

5. Experiments and Results

In this section, we introduce the experimental setup, the performance metrics, and the obtained results.

5.1. Experimental Setup. In this section, we evaluate the effectiveness of our proposed approach by collecting DNS logs from a network consisting of about 10K mobile devices for 2 weeks. This local area network with high-value information tends to be attacked by APT. Thus, there are many monitor devices deployed at the mobile base station to collect log records, including more than 300,000 DNS requests each day.

Without deploying any filters, it cannot be able to record this large volume of traffic. Hence, the volume of DNS traffic head was restored in log collector to extract DNS logs. The saved field includes source IP, destination IP, domain, query

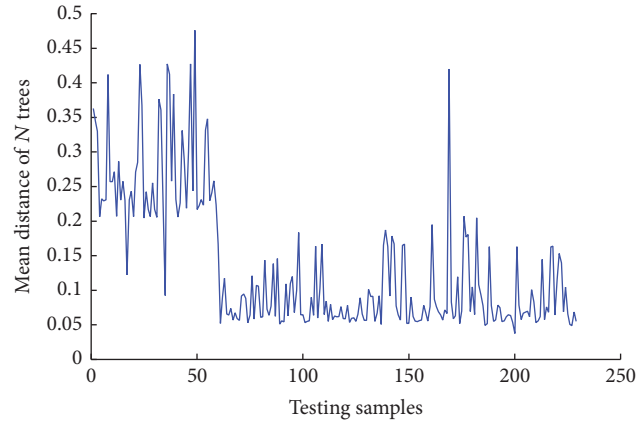


Figure 4: Difference distance between the C&C domains and normal domains.

time, and response time. The system had been implemented in Python 3.5, and all experiments were done using an off-the-shelf computer with Intel Core i7 at 3.6 GHz and 16 GB of RAM memory. In order to evaluate the true positive rates and false positive rates of our anomaly detection algorithm, we did the evaluating experiment in our training dataset including part of normal domains from the normal set and malicious domains marked by security experts.

In our experiment, the parameter $T_r = 0.2$. Almost all of malware domains' mean distance is larger than 0.2, while the mean distance of normal domains is no larger than 0.2 in our testing data. Figure 4 compares the distance between the C&C domains and normal domains. The x-axis represents different testing samples, of which the first 60 are C&C domains, and the back 170 are normal domain names. A noticeable distinction is that almost all of C&C domains' mean distance is larger than 0.2. Meanwhile, Figure 5 illustrates detection performances for malware C&C domain of different threshold. The performances of detection show our

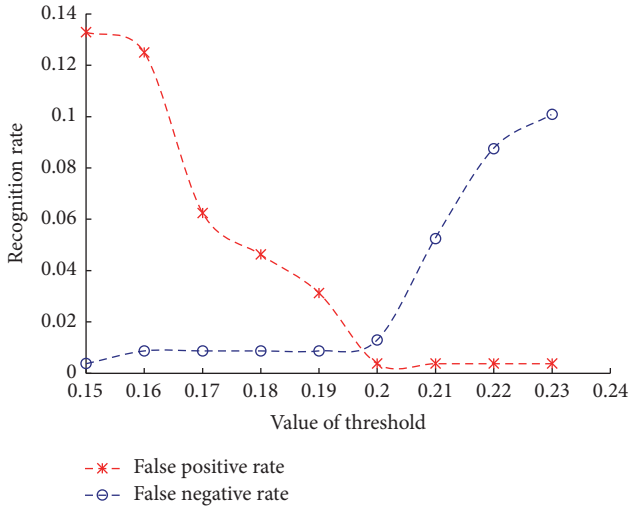


Figure 5: Recognition at different threshold.

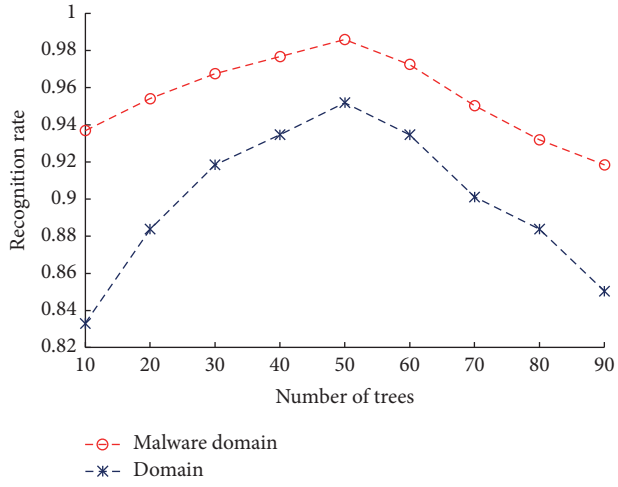


Figure 6: Recognition rate at different number of trees.

anomaly detection algorithm with the lowest false negative rate and false negative rate when the parameter $T_r = 0.2$.

Parameter $N = 50$, $M = 200$. Using the testing data, we have examined the number of trees when N increases from 10 to 90, and the number of samples when M increases from 50 to 450. The results of the experiments are presented by Figures 6 and 7. We made a statistic of recognition rate for a different number of trees and samples. As shown in Figure 6, when N increases from 10 to 50, the percentage of malicious domain identification increases; it is deduced that the scores of the number of trees are greater than 50. This is due to model overfitting. On the other hand, Figure 7 compares the effects of difference number of samples selected by each tree. Overall, when the size of samples is less than 200, false positive rate and false negative rate are decreasing. Thus, the size of samples used in each identification trees is set to 200 and the number of trees is set to 50 in our experimental environment.

The parameters are shown in Table 2.

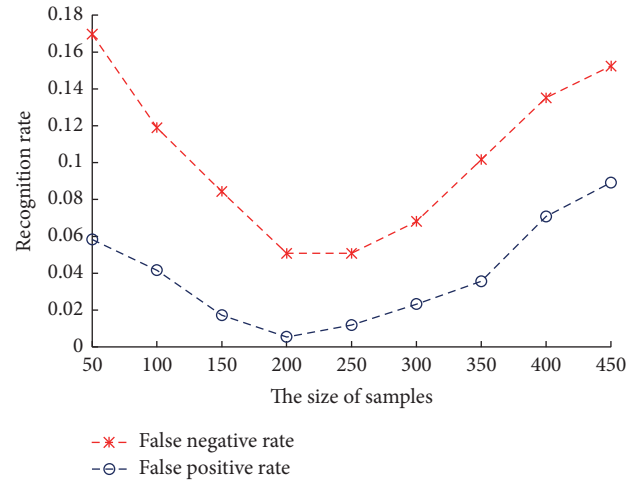


Figure 7: Recognition rate at different size of samples.

Table 2: Experimental parameters settings.

Parameter	Description	Value
T_r	Distance threshold	0.2
N	Number of trees	50
M	Number of samples	200

5.2. Results of Experiments and Discussion. The detection performances of APT malware C&C domain are expressed by performance metrics that describes both accuracy and time requirements of different detection algorithms. The accuracy is expressed by following metrics:

(1) False Recognition Rate: $\mathrm{FR} = \mathrm{FN}_n/(\mathrm{TP}_n + \mathrm{FN}_n)$

(2) Precision: $\mathrm{Pr} = \mathrm{TP}_n/(\mathrm{TP}_n + \mathrm{FP}_n)$

(3) Recall Rate: $R = \mathrm{TP}_n/(\mathrm{TP}_n + \mathrm{FN}_n)$

(4) F-Measure: $F\text{-}M = 2 \times \mathrm{Pr} \times R/(\mathrm{Pr} + R)$

In the above equations, TP_n refers to the number of normal domain names that are recognized as normal domain names, TN_n refers to the number of malicious domain names that are recognized as malicious domain names, FP_n refers to the number of malicious domain names that have been mistaken for normal domain names, and FN_n refers to the number of normal domain names that are incorrectly identified as normal domain names, respectively. Thus, the higher the value of Pr, R, and F-M, the better the recognition effect of anomaly detection algorithms. Conversely, the lower the value of FR, the better the performance.

Some experiments were performed to evaluate the performance of our proposed approach for detection APT malware C&C domains. Table 3 presents the results of different anomaly detection algorithms. GAF with information entropy yielded average detection accuracy of 98.3 percent and standard GAF yielded an average detection accuracy of 93.9 percent. Also, GAF with information entropy yielded an FP rate and FN rate of 0.013 and 0.004 percent, respectively, while standard GAF yielded an FP rate and FN rate of 0.056 and 0.004 percent, respectively. Additionally, GAF with

Table 3: Detection accuracy of different algorithms.

Algorithms	Items			
	APA	FP	FN	T (second)
iForest	0.883	0.052	0.065	17
LOF	0.765	0.169	0.109	973
KNN	0.674	0.2	0.126	4573
GAF (with information entropy)	0.983	0.013	0.004	18
GAF	0.939	0.056	0.004	15

Notes. APA, overall recognition rate; FP, false positive rate; FN, false negative rate; T, time.

Table 4: Empirical comparison of different number of trees.

Algorithms	Items			
	FR	Pr	R	F-M
iForest	0.088	0.928	0.912	0.92
LOF	0.147	0.788	0.853	0.853
KNN	0.17	0.754	0.83	0.83
GAF (with information entropy)	0.0058	0.98	0.994	0.994
GAF	0.0058	0.928	0.928	0.994

information entropy and standard GAF yielded a detection speed of 18.7 seconds and 15.6 seconds, respectively. These results revealed that the overall performance of GAF with information entropy outperformed standard GAF, implying that feature weight is a better optimization parameter.

Additionally, as shown in Table 3, GAF with information entropy was compared to three traditional anomaly detection algorithms and a detection accuracy of 98.3 percent was achieved, which is higher than the three detection accuracies (i.e., 88.3, 76.5, and 67.4 percent). Also, GAF with information entropy performed better in terms of time compared to LOF and KNN with more than 16 minutes.

Results from the experiments were compared to results of different anomaly detection algorithms. As shown in Table 4, GAF (with information entropy) has the highest PR, R, and F-M and the lowest FR. The R value of our proposed GAF algorithm reaches 0.994, which is higher than other algorithms. The F-M value and R value of GAF are higher than other three traditional algorithms. The F-M value and FR value of GAF and GAF with information entropy are the same. That was because the feature has no effect on normal sample identification. However, the PR value of GAF algorithm using differentiated information entropy to represent the weight of different features is higher than GAF whose feature has the same effect in identifying domains. Since some normal domains overlap with malware C&C domains in the feature space, LOF and KNN using all the normal samples have higher false negative rate and false positive rate. Moreover, iForest using depth of trees has certain assumptions. In our work, there are three malicious domains not yet identified since their behaviors are the same as the normal domain. The root cause of the false positives is anomaly detection.

6. Related Work

The proposed approach combines statistical knowledge related to malware using DNS to locate C&C servers with anomaly detection. Thus, the main motivation behind our work relies on APT detection, anomaly detection, DNS malicious domain detection, and botnet detection.

APT Detection. Siddiqui et al. [22] proposed a fractal based APT anomalous patterns classification method with the goal of reducing both false positives and false negatives using various features of a TCP/IP connection. Marchetti et al. [23] identified and ranked suspicious hosts possibly involved in data exfiltrations related to APT according to suspiciousness score for each internal host. Mcafee [24] extracted network features of several APT malware to identify APT C&C communication traffic. IDns [25] analyzed a large volume of DNS traffic and network traffic of suspicious malware C&C server to detect APT malware infection. Unfortunately, these approaches identified APT after data exfiltrations. Our proposed approach identifies APT malware in the stage of establishing C&C channel.

Wang et al. [26] made use of independent access to find out HTPP-based C&C domain. Barceló-Rico et al. [27] developed a semisupervised classification system to detect suspicious instances for identifying APT attacks based on HTTP traffic. However, they cannot effectively identify malware C&C domain based on other protocols. Our proposed approach uses mobile DNS logs to identify APT malware that utilizes DNS to support their C&C infrastructure.

Friedberg et al. [28] proposed an anomaly detection system to identify APT according to security logs from individual hosts. But host logs were often impractical to obtain. Bertino and Ghinita [29] detected APT related to data exfiltrations by analyzing DataBase Management System (DBMS) access logs. Liu et al. [30] made use of network traffic to identify data exfiltrations based on automatic signature generation but cannot apply even if the attacker uses encrypted communications and standard protocols. Our proposed approach identifies APT malware prior to data exfiltrations and use partial data to reduce storage overhead.

DNS Malicious Domain Detection. In order to judge whether a new domain is malicious or not, Notos [31] constructed the network, zone, and evidence-based features to compute reputation scores for new domains. However, it was dependent on large amounts of historical maliciousness data. Exposure [32] employed large-scale, passive DNS analysis techniques to detect domains that are involved in malicious activity. Unfortunately, it relied on prior knowledge of label malware C&C domain in the training phase. Notes [31] and Exposure [32] identify malicious domains based on DNS traffic from local recursive DNS servers. Unfortunately, it identified malicious domains that are misused in a variety of malicious activity. Our proposed detection approach focuses on APT malware. Other related work used graph-based inference

technique to discover new malicious domains. Manadhata et al. [33] constructed a host-domain graph to detect malicious domains combined with belief propagation. Rahbarinia et al. [34] built a machine-to-domain bipartite graph to efficiently detect new malware-control domain by tracking the DNS query behavior. Khalil et al. [35] developed graphs reflecting the global correlations among domains to discover malicious domain based on their topological connection to known malicious domains. However, those methods required prior knowledge that known partial domain names.

Botnet Detection. Botnet detection is also interesting related work to compare the problem of APT malware C&C domain detection. Sniffer [3] and BotMiner [5] detected botnet hosts based on the similarity of connections. BotGAD [4] also detected botnet from the group activity characteristics in network traffic. However, the above-mentioned detection approaches are difficult for detecting APT with limited communication samples and small-scale victims.

7. Conclusion

APT malware identification is still a challenge to network security since few attacks traces exist in mass behaviors. Most malware makes use of domain name to locate C&C server. Thus, C&C domain detection by analyzing DNS records is feasible. This paper proposes an efficient APT malware C&C domain detection approach capable of handling unmarked data. In our proposed anomaly detection algorithm, information entropy is introduced to indicate the different influence of each feature. The anomaly detector was evaluated on a dataset consisting of more than 300,000 DNS requests each day during two weeks from a mobile station. The experimental results show that our proposed approach can produce an overall R and FM coefficient of 0.994. This reveals that GAF has the highest detection accuracy rate. Moreover, our approach is applicable to the real environment without domain category.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61572115) and the Key Basic Research of Sichuan Province (Grant no. 2016JY0007).

References

[1] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings*, vol. 8735 of *Lecture Notes in Computer Science*, pp. 63–72, Springer, Berlin, Germany, 2014.

[2] M. Ask, P. Bondarenko, J. E. Rekdal et al., "Advanced Persistent Threat (APT) beyond the hype," in *Project Report in IMT4582 Network Security at GjoviN University College*, Springer, Berlin, Germany, 2013.

[3] G. Gu, J. Zhang, and W. Lee, "BotSniffer: detecting botnet command and control channels in network traffic," in *Proceedings of the 15th Annual Network and Distributed System Security Symposium*, 2008.

[4] H. Choi, H. Lee, and H. Kim, "BotGAD: detecting botnets by capturing group activities in network traffic," in *Proceedings of the 4th International ICST Conference on Communication System Software and Middleware (COMSWARE '09)*, June 2009.

[5] G. Gu, R. Perdisci, J. Zhang et al., "BotMiner: clustering analysis of network traffic for protocol-and structure-independent botnet detection," *USENIX Security Symposium*, vol. 5, no. 2, pp. 139–154, 2008.

[6] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware C8C detection," *ACM Computing Surveys*, vol. 49, no. 3, article 59, 2016.

[7] K. Zetter, "Google hack attack was ultra sophisticated, new details show," *Wired Magazine*, vol. 14, 2010.

[8] Y. Zhang, C. Xu, H. Li, and X. Liang, "Cryptographic public verification of data integrity for cloud storage systems," *IEEE Cloud Computing*, vol. 3, no. 5, pp. 44–52, 2016.

[9] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 159–170, 2015.

[10] R. Sonawane, T. Tajane, P. Chavan et al., "Anomaly based intrusion detection network system," *Software Engineering and Technology*, vol. 8, no. 3, pp. 66–69, 2016.

[11] M. Wan, L. Li, J. Xiao, C. Wang, and Y. Yang, "Data clustering using bacterial foraging optimization," *Journal of Intelligent Information Systems*, vol. 38, no. 2, pp. 321–341, 2012.

[12] X. Liu, X. Zhang, Y. Jiang, and Q. Zhu, "Modified t-distribution evolutionary algorithm for dynamic deployment of wireless sensor networks," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 6, pp. 1595–1602, 2016.

[13] N. Suryavanshi and A. Jain, "Phishing detection in selected feature using modified SVM-PSO," *International Journal of Research in Computer and Communication Technology*, vol. 5, no. 4, pp. 208–214, 2016.

[14] W. Wang, L. Li, H. Peng, J. Xiao, and Y. Yang, "Synchronization control of memristor-based recurrent neural networks with perturbations," *Neural Networks*, vol. 53, pp. 8–14, 2014.

[15] M. X. Ma, H. Y. Ngan, and W. Liu, "Density-based outlier detection by local outlier factor on largescale traffic data," *Electronic Imaging*, vol. 2016, no. 14, pp. 1–4, 2016.

[16] J. A. Khan and N. Jain, "Improving intrusion detection system based on KNN and KNN-DS with detection of U2R, R2L attack for network probe attack detection," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 5, pp. 209–212, 2016.

[17] L. Sun, S. Versteeg, S. Boztas, and A. Rao, "Detecting anomalous user behavior using an extended isolation forest algorithm: an enterprise case study," https://arxiv.org/abs/1609.06676.

[18] P. Singh Chahal and S. Singh Khurana, "TempR: application of stricture dependent intelligent classifier for fast flux domain detection," *International Journal of Computer Network and Information Security*, vol. 8, no. 10, pp. 37–44, 2016.

[19] A. R. Kang, J. Spaulding, and A. Mohaisen, "Domain name system security and privacy: old problems and new challenges," https://arxiv.org/abs/1606.07080.

[20] B. Yu, L. Smith, and M. Threefoot, "Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic," in *Machine Learning and Data Mining in Pattern Recognition: 10th International Conference, MLDM 2014, St. Petersburg, Russia, July 21–24, 2014. Proceedings*, vol. 8556 of *Lecture Notes in Computer Science*, pp. 258–271, Springer International Publishing, 2014.

[21] Alexa Web Information Company, 2015, http://www.alexa.com/topsites.

[22] S. Siddiqui, M. S. Khan, K. Ferens, and W. Kinsner, "Detecting advanced persistent threats using fractal dimension based machine learning classification," in *Proceedings of the 2nd ACM International Workshop on Security and Privacy Analytics (IWSPA '16)*, pp. 64–69, ACM, 2016.

[23] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for Advanced Persistent Threat detection," *Computer Networks*, vol. 109, pp. 127–141, 2016.

[24] N. Villeneuve and J. Bennett, *Detecting Apt Activity with Network Traffic Analysis*, Trend Micro Incorporated, 2012, http://www.trendmicro.pl/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf.

[25] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[26] X. Wang, K. Zheng, X. Niu, B. Wu, and C. Wu, "Detection of command and control in advanced persistent threat based on independent access," in *Proceedings of the IEEE International Conference on Communications (ICC '16)*, pp. 1–6, Kuala Lumpur, Malaysia, May 2016.

[27] F. Barceló-Rico, A. I. Esparcia-Alcázar, and A. Villalón-Huerta, "Semi-supervised classification system for the detection of advanced persistent threats," in *Recent Advances in Computational Intelligence in Defense and Security*, pp. 225–248, Springer International Publishing, 2016.

[28] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: from network event correlation to incident detection," *Computers and Security*, vol. 48, pp. 35–57, 2015.

[29] E. Bertino and G. Ghinita, "Towards mechanisms for detection and prevention of data exfiltration by insiders," in *Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pp. 10–19, ACM, March 2011.

[30] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, "SIDD: a framework for detecting sensitive data exfiltration by an insider attack," in *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences (HICSS '09)*, pp. 1–10, January 2009.

[31] M. Antonakakis, R. Perdisci, D. Dagon et al., *Notos: Building a Dynamic Reputation System for DNS*, Georgia Institute of Technology College of Computing, Atlanta, Ga, USA, 2010.

[32] L. Bilge, E. Kirda, C. Kruegel et al., "EXPOSURE: finding malicious domains using passive DNS analysis," in *Proceedings of the Annual Network and Distributed System Security Symposium (NDSS '11)*, 2011.

[33] P. K. Manadhata, S. Yadav, P. Rao, and W. Horne, "Detecting malicious domains via graph inference," in *Computer Security—ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014. Proceedings, Part I*, vol. 8712 of *Lecture Notes in Computer Science*, pp. 1–18, Springer International Publishing, 2014.

[34] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: efficient behavior-based tracking of malware-control domains in large ISP networks," in *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '15)*, pp. 403–414, June 2015.

[35] I. Khalil, T. Yu, and B. Guan, "Discovering malicious domains through passive DNS data graph analysis," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security (ASIA CCS '16)*, pp. 663–674, ACM, June 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 3247627, 8 pages
https://doi.org/10.1155/2017/3247627

Research Article

A Stable-Matching-Based User Linking Method with User Preference Order

Xuzhong Wang, Yan Liu, and Yu Nan

China State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China

Correspondence should be addressed to Yan Liu; ms_liuyan@aliyun.com

Received 10 December 2016; Revised 18 February 2017; Accepted 1 March 2017; Published 28 March 2017

Academic Editor: Zonghua Zhang

With the development of social networks, more and more users choose to use multiple accounts from different networks to meet their needs. Linking a particular user's multiple accounts not only can improve user's experience of the net-services such as recommender system, but also plays a significant role in network security. However, multiple accounts of the same user are often not directly linked to each other, and further, the privacy policy provided by the service provider makes it harder to find accounts for a particular user. In this paper, we propose a stable-matching-based method with user preference order for the problem of low accuracy of user linking in cross-media sparse data. Different from the traditional way which just calculates the similarity of accounts, we take full account of the mutual influence among multiple accounts by regarding different networks as bilateral (multilateral) market and user linking as a stable matching problem in such a market. Based on the combination of Game-Theoretic Machine Learning and Pairwise, a novel user linking method has been proposed. The experiment shows that our method has a 21.6% improvement in accuracy compared with the traditional linking method and a further increase of about 7.8% after adding the prior knowledge.

1. Introduction

Over the past decade, followed by the exponentially growing net-services, the number of anonymous users is also springing up. As of the third quarter of 2016, active users of Facebook reached 1.79 billion [1], which means more than half of 30 million Internet users use Facebook per month at least once. About 65% or about 1.18 billion users log at least once in daily. However, some traditional social network sites now are facing significant development. According to the Twitter 2016 Q3 results [2], average growth rate of monthly active users, only about 3%, reached 317 million, compared with the image-based social network Instagram, whose monthly active users have already exceeded 600 million [3]. This change shows that, with the development of times, user's interest of the net-services has been divided. Therefore, net-services providers also aim at developing different social services for various user's interests.

Nowadays, each net-service often has its unique mode of information sharing to maintain its social relationships. These unique models attract different user groups; for example, a user selects Twitter to share some information publicly and chooses Facebook for own circles, and for sharing traveling scenery and foods, of course Instagram is the best choice. On these net-services platforms, users typically pass a uniquely identified nickname along with some other attribute tags, such as profile information, hobbies, friendships, and events. If these accounts can be effectively linked with a particular user, when we try to understand a user comprehensively, this not only can significantly improve his (or her) experience of a recommender system but also can provide a better anonymity protection policy [4]. In network security, when detecting malicious attackers with multiple accounts in different platforms, it is possible to integrate the information of cross-media together and makes a vast improvement of the detecting ability. Practice has proved user linking has important practical significance.

However, due to the anonymity protection policy of net-services providers and users in different net-service platforms always choose to share different information, resulting in the fact that particular user's multiple accounts often do not have

adamant relevance. This large number of nondirectly linked account makes difficulties for comprehensively understanding the user. There are existing studies done by analyzing the user's naming style convention [5, 6], profile [7, 8], writing style [9], behavior [10], social relations [11, 12], and so on and then by linking users multiple accounts by statistical and also machine learning methods. These methods are used to model the characteristics of a vast number of accounts and made some certain achievements in the experimental dataset. However, in reality, not enough account features could be gotten from sparse network data and the behavior behind these accounts is always changing. It is hard to use a stable mathematical model to describe it. Moreover, the real human behavior is neither random nor entirely rational [13]. Therefore, considering the mutual influence among multiple accounts from different net-services users, the user linking problem can be regarded as a cooperative game problem in the bilateral (multilateral) market—how to formulate a cooperation (linking) strategy in the markets (net-services) to enhance the interest (linking results) of the whole candidate accounts set.

In recent years, researches on Game-Theoretic Machine Learning are progressing; some researchers have constructed a Game-Theoretic Machine Learning framework, through the Markov model to study and predict the user's behavior [13–15]; some scientists use cooperative game approach to evaluate and select the features for machine learning [16]. These methods have proved the game theory plays an improving role on the traditional machine learning. Therefore, this paper proposes a stable-matching-based game theory method for user linking with user preference order and prior knowledge. The main contributions of this paper are as follows.

Process a novel method based on stable matching game theory to carry on the analysis of user linking.

Input the linked user accounts as prior knowledge to enhance the result of user linking.

Through many experiments carried out in the LifeSpec [17] project dataset provided by Microsoft Research Asia, our method is about 21.6% higher in accuracy compared with the traditional user linking methods. Moreover, there was a further improvement of about 7.8% after inputting the prior knowledge.

2. Problem Formulation

In this section, the related concepts and formal descriptions of user linking are given. For the convenience of description, this paper focuses on two heterogeneous networks. Symbols used in this paper are shown in Table 1.

User. A user can be represented as $u_i = \{a_i^1, a_i^2, a_i^3, \ldots, a_i^n\}$, where $a_i^k = \{f_1, f_2, \ldots, f_p\}$ represents the user u_i's account on the network k and f represents feature of accounts. For convenience, we focus on two heterogeneous networks, so $u_i = \{a_i^s, a_i^t\}$, where a_i^s represents the account from the source network s and a_i^t represents the account from the target network t.

Table 1: Symbols used in this paper.

Symbols	Significance
u	User
a	User account
A	Account set
pair	Accounts pair
l_{pair}	Identification of accounts pair
s	Source network
t	Target network

Account Set. An account set represents extractable accounts from a particular network. So $A^s = \{a_1^s, a_2^s, a_3^s, \ldots, a_p^s\}$ represents source network account set and $A^t = \{a_1^t, a_2^t, a_3^t, \ldots, a_q^t\}$ represents target network account set, where p, q are the number of users of both networks.

Accounts Pair. Accounts pair pair $= (a_m^s, a_n^t)$ represents a tuple consisting of any account a_m^s of user u_m from the source network s and any account a_n^t of user u_n from the target network t.

Identification of Accounts Pair. $l_{\text{pair}=(a_m^s, a_n^t)} = \{0, u_m \neq u_n;\ 1, u_m = u_n\}$, which means when the accounts pair consisted of accounts from the same user, the value of identification $l_{\text{pair}} = 1$; otherwise $l_{\text{pair}} = 0$.

Problem Description. Given the source network s and target network t, extracting the candidate account sets A^s, A^t, and grouping any two accounts from these networks one by one to an accounts pair, then get $n * m$ pairs. Finally, use a linking algorithm to find all the pairs whose identification $l_{\text{pair}} = 1$, namely, linking accounts $a_i^s,\ a_j^t$ from two heterogeneous networks $s,\ t$.

The challenges of this paper are as follows:

(1) Traditional user linking technology is often trying to maximize some objective function so that the whole candidate accounts set can get the best result. However, since the user's different account behavior is often not rational and stable [13] and the sparse features of accounts could influence the linking result significantly, the traditional methods do not always have an ideal result on large-scale sparse data sets. Within the cooperative game theory, user linking is actually trying to find matched players in the bilateral market. In this paper, we combined the game theory and the user's preference using stable matching theory [18] and Pairwise, finally linking users through the cooperation between accounts.

(2) The traditional method often linked the user's account by calculating the "similarity" between different accounts using certain types characteristics. However, in the real world, multiple accounts of a user on various platforms tend to reflect different needs of the user, resulting in the fact that the "similarity" is

minuscule that many accounts can not be linked. Taking into account the fact that the "user linking problem" and "linking similar user problem" are different, so we input some linked user as prior knowledge, thereby enhancing the result of user linking.

The following section will detail how to solve these two problems.

3. Stable User Linking with User Preference Order

User linking essentially is a multiclassification problem, and different user accounts are categorized according to the user category. However, because the multiclassification problem is usually difficult to obtain an ideal solution, therefore, in this paper, we make use of the idea of Pairwise [19], combined accounts to pairs, and classified them according to whether linked. Then user linking problem will be converted into a binary classification problem and could calculate the probability of each account pair under a different category. Then, according to this probability, construct the user preference order set and finally convert the question into "how to select the best target account in one's preference order set" and try to improve it by inputting the prior knowledge. Therefore, we present a three-phase approach to solve the user linking problem:

(1) Constructing user preference order set: calculating posterior probability P for each pair according to the SVM model trained by the training set and sorting P of each a^s to construct user preference order set.

(2) User linking based on stable matching: using stable matching algorithm based on the user preference order set between A^s, A^t and finally getting all the stable links among accounts.

(3) User linking based on prior knowledge: inputting the prior knowledge to improve user linking in the stable matching algorithm and finally get the reinforced user linking algorithm.

3.1. Constructing User Preference Order Set. According to Pairwise, first user linking can be converted into a binary classification problem, and by calculating the classification probability the account preference order set could be constructed, which is defined as follows.

User Preference Order Set. For an account a^s, the ordered sequence $\{a_1^t > a_2^t > a_3^t > \cdots > a_n^t\}$ of the target account set A^t is called the preference order set of the account a^s. The ordered sequence reflects the order of which target account a^s is more likely to link.

In recent years, many kinds of research have shown that the Support Vector Machine has a high ability in resolving the problem of binary classification [20, 21]. Since SVM is very sensitive to features, selecting the proper feature is vital. Traditional methods make many features by artificial information, such as naming habits, personal profiles, writing style, user behavior trajectory, and social relations. However, due to the incompleteness and heterogeneity of network data, the features of user data acquired not only can be very limited but also need to be completed. Therefore, by using account labels, we avoid the difficulty of filtering and completing of features.

From the reality net-services, some of them provide labels to simply and clearly reflect the characteristics of user accounts. But others do not have. So, we can directly construct labels by account history text using topic model, such as LDA. The method of label extraction by the topic model has been matured in recent years and will not be repeated here.

In this paper we took these accounts with their label tag as a bag-of-words model and then calculate the value of features between a_m^s's feature vector $T_m^s = \{\text{tag}_1, \text{tag}_2, \ldots, \text{tag}_p\}$ and a_n^t's feature vector $T_n^t = \{\text{tag}_1', \text{tag}_2', \ldots, \text{tag}_q'\}$ as follows:

(1) Cosine similarity: $\cos = (T_m^s \cdot T_n^t)/\|T_m^s \times T_n^t\|$.

(2) Number of common labels: $n = \#(T_m^s \cap T_n^t)$.

According to the feature above, the training data can be trained by SVM and then accurately classify the test data. However, in large-scale data, there are many accounts because of the sparseness of labels and the different user's accounts may have some similarity, resulting in the fact that many cases can not make an accurate classification. These noise accounts will have a great impact on the classifying effect when using the standard SVM. In fact, user linking is a nondeterministic classification problem: some samples can not belong to a category accurately, only through the probability to reflect its belonging to a certain category. To address this issue, according to the sigmoid-fitting method proposed by Platt [22], we calculate each pair's posterior probability P under the conditions $l_{\text{pair}} = 1$:

$$P\left(l_{\text{pair}} = 1 \mid f\right) = \frac{1}{1 + e^{Af+B}}, \tag{1}$$

where f is the Support Vector Machine with no threshold output $f(x) = w^t x + b$ and two parameters A, B can be set by maximum likelihood estimation of the training set. This posterior probability actually reflects the likelihood that one account will be linked to another target account. According to the posterior probability we construct user preference order set as follows.

Based on Pairwise, the training set and the test set of pairs are constructed between account sets A^s, A^t, and the feature vectors of any pairs are constructed by using the above two features, and then use Support Vector Machine to train a model on the training set. For a particular test set account a^s, calculate the posterior probability P of pair $= (a^s, a^t)$, where a^t comes from target network t, under the conditions $l_{\text{pair}} = 1$. Finally, we get the user preference order set $\{a_1^t > a_2^t > a_3^t > \cdots > a_n^t\}$ of a^s by sorting P of each pair.

The following section describes how to link user accounts by user preference order set.

Input: account set A^s, A^t
Output: result set T
(1) Initializes the result set $T = \emptyset$
(2) Calculate the posterior probability P of any pair = (a_s, a_t)
(3) Sort to get the preference order set for each account
(4) **if** $A^s \cdot \text{length}() \neq A^t \cdot \text{length}()$ **then**
(5) Add $A^s \cdot \text{length}() - A^t \cdot \text{length}()$ fake accounts to the small parties, and set the preference order set keeps empty
(6) **end if**
(7) **while** Exists any account $a_s \in A^s$ is not linked && a_s's preference order set $\neq \emptyset$ **do**
(8) Find the most preferred target account a_t from a_s's preference order set and remove it
(9) **if** a_t is not linked && a_s is in the preference order of a_t **then**
(10) Set a_s, a_t linked, $T = T \cup \{(a_s, a_t)\}$
(11) **else if** a_t is linked **then**
(12) Get the linking object a_m of a_t
(13) **if** a_s is in the preference order set of a_t && $a_s > a_m$ **then**
(14) Cancel the linking state of a_m, $T = T - \{(a_m, a_t)\}$
(15) Set a_s, a_t linked, $T = T \cup \{(a_s, a_t)\}$
(16) **end if**
(17) **end if**
(18) **end while**
(19) Remove all the accounts linked with a_f
(20) **return** T

Algorithm 1: Stable User Linking with Preference order, SULP.

3.2. User Linking Based on Stable Matching. Through the convention in Section 3.1, user linking actually turns into "how to select the best target account in one's preference order set" so that the whole candidate account set can get the best performance. In this paper, we try to use stable matching theory to solve this problem. The stable matching theory [18] is proposed by Shapley using cooperative game theory to solve the linking problem in bilateral market entities. Because of this theory, Shapley won the 2012 Nobel Prize in Economics. This theory has been widely used in many practical scenarios, such as students selecting (students and schools matching [23]), housing allocation (matching between people and house [24]), and job searching (employee and employer matching [25]). The core of this theory lies in the realization of the stable state, which means there does NOT exist ANY pair of entities in the bilateral market at the end of linking, which have a more preferred target than the currently linking target. In fact, if the source network s and target network t are regarded as a bilateral market, user accounts a_m^s, a_n^t can be seen as entities from the bilateral market. Then the problem of "how to select the best target account in one's preference order set" is converted to "how to find a cooperation (linking) strategy in the markets (networks) to make the interest (linking results) to the maximum." Therefore, based on the idea of stable matching, we linked accounts based on the preference order set.

Broken Account Pair. If an account a_m^s is linking to $a_{m'}^t$, a_n^t is linking to $a_{n'}^s$. Assume there is a pair = (a^s, a_n^t) on which the account a_m^s has $a_n^t > a_{m'}^t$ in its preference order set and the account a_n^t has $a_m^s > a_{n'}^s$ in its preference order set; then the pair = (a^s, a_n^t) is called a broken account pair because actually it breaks the current linked pairs.

Stable Matching. If there does NOT exist ANY broken account pair at the end of linking, then we said the entire linking is a stable matching.

Using [18] proposed GS delay algorithm can achieve a stable matching in the bilateral market. However, the standard GS algorithm requires that the number of entities in the bilateral market must be N, and the size of preference order set of each entity must also be the size N. That is to say, "the number of bilateral market entities is same" and "each preference order set is completed." However, these two restrictions are difficult to meet, and because of the lack of attributes, some of the feature vectors can not be calculated and can not get the completed order set, so we make two adaptations.

(1) Fake account: an account which does NOT actually exist is called fake account a_f. In a linking process, a balanced number between two account sets of fake accounts will be added to the littler set, and when linking is completed all the pairs which contain fake account will be excluded.

(2) Uncompleted user preference order set: a user preference order set which does NOT include ALL the accounts in the target network is called an uncompleted user preference order set. In a linking process, if a^t is not in a^s's user preference order set we directly denied this link.

According to this, we propose a stable-matching-based user linking method with user preference order (Stable User Linking with Preference order, SULP) as shown in Algorithm 1.

Through Algorithm 1, this paper combines the user preference order and stable matching of cooperative game

(1) the same as SULP Algorithm 1 line 1–8
(2) **if** a_t is not linked && a_s is in the preference order of a_t **then**
(3) Set a_s, a_t linked, $T = T \cup \{(a_s, a_t)\}$
(4) **else if** a_t is linked **then**
(5) Get the linking object a_m of a_t
(6) **if** a_s is a prior candidate account of a_t **then**
(7) **if** a_m is not a prior candidate account of $a_t \parallel a_s < a_m$ **then**
(8) Cancel the linking state of a_m, $T = T - \{(a_m, a_t)\}$
(9) Set a_s, a_t linked, $T = T \cup \{(a_s, a_t)\}$
(10) Clear all the accounts behind a_s's preference order list
(11) **end if**
(12) **end if**
(13) **end if**
(14) Remove all the accounts linked with a_f
(15) **return** T

Algorithm 2: EXtended Stable User Linking with Preference order, EXSULP.

theory to achieve the purpose of user linking. The next section will be on how to strengthen the result of this method.

3.3. User Linking Based on Prior Knowledge. Consistent with the traditional linking method, the method we proposed is still based on the similarity of account features. However, in fact, as the network platform tends to specify functionally, users on different platforms usually choose to explicitly express their interest by their multiple accounts, and these various interests among the accounts are likely to have little similarity. Therefore, user linking not only is "how to link accounts by similarity," but also includes "how to identify and link the accounts which are dissimilar but belong to the same user." The latter one is extremely challenging, and the researches show that there has been no effective solution. In this paper, we try to input some users' linked accounts as prior knowledge, to strengthen the user linking method proposed in Section 3.2.

Considering that the preference order set of the entity in the bilateral market is a set based on the feature similarity, the above method can not adequately reflect the correlation information among different accounts. To add some correlative information by prior knowledge, we defined prior candidate account set as follows.

Prior Candidate Account Set. For an account a^s, given its linked account a^t, then a^t is called a prior candidate account of a^s. In the matching process, a^s is assumed to match account $a^{t'}$, if $a^{t'}$ is NOT a prior candidate account of a^s; then regardless of the preference order set, let a^t link to a^s. If $a^{t'}$ IS a prior candidate account of a^s, then follow the order of preference set.

Based on the definition above, we further propose a reinforced algorithm (EXtended Stable User Linking with Preference order, EXSULP) based on prior knowledge. Only the improved part is shown in Algorithm 2.

According to the algorithm, we input the already linked account as the prior knowledge, further strengthening the possible correlation between the accounts. Finally, all the eligible pairs $= (a^s_m, a^t_n)$ are taken as the final result of user linking between the network s and network t.

Table 2: My second table.

Dataset	Number of accounts	Number of comments	Number of works
Books	34942	2118400	523064
Movies	41823	8397846	82868

4. Experiments

In this section, based on the dataset provided by Microsoft Research Asia LifeSpec [17], we used the standard SVM, SVM based on the cooperative game theory, and reinforced SVM based on prior knowledge, respectively, to analyze user linking. Experiment code has been made public on GitHub: https://github.com/Observerspy/UserStableMatching.

4.1. Dataset Description. LifeSpec is a computational framework developed by the Microsoft Research Asia for discovering and hierarchically categorizing urban lifestyles. The LifeSpec dataset is composed of tens of millions of user's data about sign-in, movie comments, book comments, music comments, and behavior. In this paper, we attempt to link users from the books set as the source network s and to movies set as the target network t.

As in Table 2, we selected a total of 62,558 different users.

(1) Books Dataset: contains 34,942 different accounts on 523,064 books with 2,118,400 comments; each data contains title, author, publisher, date of issue, number of pages, price, packaging, labels, user ratings, and other information.

(2) Movies Dataset: contains 41,823 different accounts on 82,868 movies with 8,397,846 comments; each data contains name, director, screenwriter, starring,

category, country, duration, release date, labels, user ratings, and other information.

The total number of pairs in this dataset is 1,461,379,266. Because, in such a large-scale dataset, the proportion of positive instances and negative instances is often more than 1 : 10000, we controlled the proportion to about 1 : 1 by random undersampling.

4.2. Performance of User Linking Methods. We took labels from the books and movies as the accounts features and the frequency of each label as the feature value. Because the dimension of inputting feature vector is small, we use ten times 10-fold cross-validation Gaussian kernel SVM with setting the cost value to 1 and remaining the default parameters. Support Vector Machines and posterior probability calculations are provided by LibSVM [26] tools. The compared methods are summarized as follows.

(1) SVM_Label: baseline method, using SVM to do a link\nonlink classification only in label feature space.

(2) SULP: the stable-matching-based user linking method with user preference order which is proposed in Section 3.2.

(3) EXSULP: the extended user linking method which is proposed in Section 3.3.

As the user linking problem only concerned with the correct links (positive instances), therefore, we select the precision p, recall r, and $F1$ value $F1$ as the evaluation metrics, and the average result of 10 times 10-fold cross-validation is shown in Table 3.

It can be seen from the results that the two methods proposed in this paper have surpassed the baseline method on the metrics of precision p, recall r, and $F1$, where the SULP has an improvement of about 21.6% in accuracy and a further increase of about 7.8% after adding the prior knowledge. Compared with other researches which used a large number of user's personal information, texts, behaviors, and so on, we achieved the ideal precision when only using the labels as a feature. Moreover, different from other stable matching methods [27], we canceled the two restriction conditions of the following: "the number of bilateral market entities must be same" and "the preference order set is completed." Therefore, in the complex sparse real dataset, the method proposed in this paper can be considered to have better practical significance.

4.3. Analysis of Prior Knowledge. From the experiment above, we can know that the prior knowledge can improve the performance of user linking. It is clear that the proportion of prior knowledge to the whole data will influence the final linking results. Therefore, we analyze EXSULP algorithm by taking a part of incorrect classification results (total 2158) obtained from SULP algorithm as a prior knowledge and changing the proportion of the prior knowledge to analyze the effect of prior knowledge.

Expansion Rate

$$\text{Expansion rate} = \frac{\#\left(\text{EXSULP right classification}\right) - \#\left(\text{SULP right classification}\right)}{\text{a priori knowledge's proportion of total incorrect instances}}, \tag{2}$$

representing the extended ability of the EXSULP algorithm for linking results.

The result is shown in Figure 1.

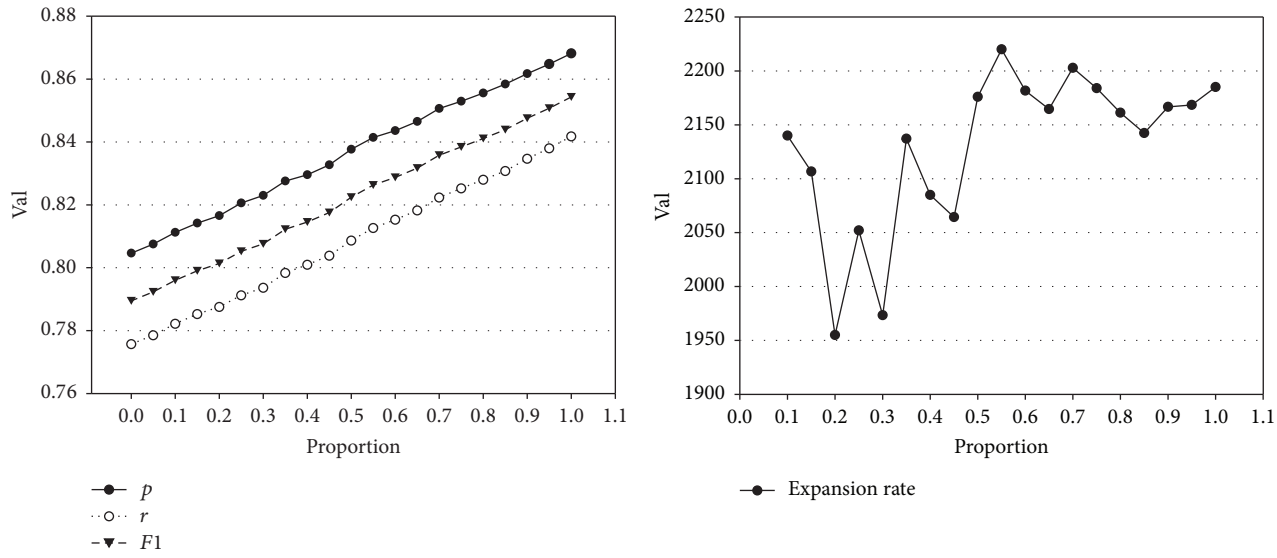
From the results, it can be seen with the increasing proportion that p, r, and $F1$ values increase steadily. It can be considered the proportion of prior knowledge is in proportion to the result of the algorithm, enhancing the precision of up to about 7.8%. The expansion rate reflects the fact that the results of this algorithm gradually stabilize as the scale of prior knowledge increases. The above experiment sufficiently proved the prior knowledge can enhance the correlation among accounts, illustrating the effectiveness of our method.

4.4. Case Study. We choose four linked results to display and analyze in Table 4. The coexisting top 10 labels are given (translated to English, the works name is in italic), among which 1–3 are the correct links and 4 is a wrong link.

As can be seen from Table 4, because of the semantics of the label, when the coexisting labels are specific enough, then the accounts can be correctly linked. In fact, it further illustrates that the problem of user linking can be solved according to each user's specific and unique interest labels. However, as shown in item 4, when these labels represent more abstract and general terms, these accounts can not make the right link. When the label of the linked account which is inputted as the prior knowledge contains such abstract and general terms, it can effectively reduce the misclassification caused by the classifier based on calculating features.

5. Conclusion

In this paper, we have studied the user linking problem and propose a stable-matching-based method with user preference order. Different from the restrictions of the traditional stable matching algorithm, we made some relaxation and enhance the result of user linking by inputting prior knowledge. Experiments show that, in the real dataset, our method has achieved an ideal effect when only using the characteristics of the website label, which adequately demonstrates the effectiveness of this approach. In the future research, we will further study how to extract accurate and efficient characteristics in the sparse data and how to enhance the correlation between different accounts.



(a) Effect on the p, r, $F1$ values

(b) Effect on the expansion rate

Figure 1: Effect of CGSVMEX on the proportion of prior knowledge.

Table 3: Performance of different methods.

Methods	p	r	$F1$
EXSULP	**86.8%**	**84.2%**	**85.5%**
SULP	80.5%	77.6%	79.0%
SVM_Label	66.2%	40.9%	50.6%

Table 4: Four linked instances.

ID	Coexisting labels
1	Japan, Love, Animation, Magic, Youth, *Suzumiya Haruhi, Higashino Keigo, Makoto Shinkai, Da Vinci Code, Harry Potter*
2	American, Japan, Love, Classic, China, *Perfume, Pride and Prejudice, Shunji Iwai, Mayday, Fairy Tale*
3	Japan, Childhood, British, Magic, *Dragon Ball, Saint Seiya, Slam Dunk, Doraemon, Harry Potter, Garfield*
4	American, Love, British, Humanity, Hong Kong, Science Fiction, China, Youth, Growth, Japan

Conflicts of Interest

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

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grant nos. 61309007, U1636219) and the National Key Research and Development Program of China (Grant no. 2016YFB0801303). The dataset is provided by Microsoft Research Asia.

References

[1] Facebook Inc, "Facebook quarterly report [sections 13 or 15(d)]," https://www.sec.gov/Archives/edgar/data/1326801/000132680116000087/0001326801-16-000087-index.htm.

[2] Twitter, Twitter quarterly report [sections 13 or 15(d)], https://www.sec.gov/Archives/edgar/data/1418091/000156459016026749/0001564590-16-026749-index.htm.

[3] I. Inc, "600 Million and counting".

[4] T. Ma, J. Zhou, M. Tang et al., "Social network and tag sources based augmenting collaborative recommender system," *IEICE Transactions on Information and Systems*, vol. E98-D, no. 4, pp. 902–910, 2015.

[5] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media (ICWSM '09)*, pp. 354–357, San Jose, Calif, USA, May 2009.

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

[7] J. Vosecky, D. Hong, and V. Y. Shen, "User identification across multiple social networks," in *Proceedings of the 1st International Conference on Networked Digital Technologies (NDT '09)*, pp. 360–365, IEEE, Ostrava, Czech Republic, July 2009.

[8] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischofi, "Identifying users across social tagging systems," in *Proceedings of the 5th Annual Conference on Weblogs and Social Media (ICWSM '11)*, Barcelona, Spain, July 2011.

[9] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," *Digital Investigation*, vol. 7, no. 1-2, pp. 56–64, 2010.

[10] Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie, "You are where you go: inferring demographic attributes from location check-ins," in *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM '15)*, pp. 295–304, ACM, Shanghai, China, February 2015.

[11] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '08)*, pp. 111–125, IEEE, Oakland, Calif, USA, May 2008.

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

[13] H. Li, F. Tian, W. Chen, T. Qin, and T.-Y. Liu, "Generalization analysis for game-theoretic machine learning," https://arxiv.org/abs/1410.3341.

[14] F. Tian, H. Li, W. Chen, T. Qin, E. Chen, and T.-Y. Liu, "Agent behavior prediction and its generalization analysis," https://arxiv.org/abs/1404.4960.

[15] D. He, W. Chen, L. Wang, and T.-Y. Liu, "A game-theoretic machine learning approach for revenue maximization in sponsored search," https://arxiv.org/abs/1406.0728.

[16] X. Sun, Y. Liu, J. Li, J. Zhu, H. Chen, and X. Liu, "Feature evaluation and selection with cooperative game theory," *Pattern Recognition*, vol. 45, no. 8, pp. 2992–3002, 2012.

[17] N. J. Yuan, F. Zhang, D. Lian, K. Zheng, S. Yu, and X. Xie, "We know how you live: exploring the spectrum of urban lifestyles," in *Proceedings of the 1st ACM Conference on Online Social Networks (COSN '13)*, pp. 3–14, ACM, Boston, Mass, USA, 2013.

[18] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[19] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[20] B. Gu and V. S. Sheng, "A robust regularization path algorithm for ν-support vector classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

[21] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403–1416, 2015.

[22] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[23] Y. Chen and O. Kesten, "From boston to shanghai to deferred acceptance: theory and experiments on a family of school choice mechanisms in," in *Proceedings of the International Conference on Auctions, Market Mechanisms and Their Applications*, pp. 58–59, Springer, New York, NY, USA, April 2011.

[24] P. Guillen and O. Kesten, On-campus housing: theory vs. experiment.

[25] A. E. Roth and E. Peranson, "The effects of the change in the NRMP matching algorithm," *The Journal of the American Medical Association*, vol. 278, no. 9, pp. 729–732, 1997.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article no. 27, 2011.

[27] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 179–188, ACM, San Francisco, Calif, USA, 2013.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8158465, 8 pages
https://doi.org/10.1155/2017/8158465

Research Article

Semitensor Product Compressive Sensing for Big Data Transmission in Wireless Sensor Networks

Haipeng Peng,[1,2] Ye Tian,[1,2] and Jürgen Kurths[3]

[1]*Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*Potsdam Institute for Climate Impact Research, Potsdam 14473, Germany*

Correspondence should be addressed to Haipeng Peng; penghaipeng@bupt.edu.cn

Received 16 January 2017; Accepted 9 March 2017; Published 22 March 2017

Academic Editor: Liu Yuhong

Big data transmission in wireless sensor network (WSN) consumes energy while the node in WSN is energy-limited, and the data transmitted needs to be encrypted resulting from the ease of being eavesdropped in WSN links. Compressive sensing (CS) can encrypt data and reduce the data volume to solve these two problems. However, the nodes in WSNs are not only energy-limited, but also storage and calculation resource-constrained. The traditional CS uses the measurement matrix as the secret key, which consumes a huge storage space. Moreover, the calculation cost of the traditional CS is large. In this paper, semitensor product compressive sensing (STP-CS) is proposed, which reduces the size of the secret key to save the storage space by breaking through the dimension match restriction of the matrix multiplication and decreases the calculation amount to save the calculation resource. Simulation results show that STP-CS encryption can achieve better performances of saving storage and calculation resources compared with the traditional CS encryption.

1. Introduction

For its ease of deployment and cost effectiveness, wireless sensor network (WSN) is widely used in environment monitoring, disaster relief, military, and so on [1–3]. For example, with WSN for forest fire monitoring, numerous sensors are deployed in the monitor area, and big data should be gathered and transmitted in real-time. Big data transmission consumes vast energy, but the node in WSNs is energy-limited. If the battery of the node is drained, the node is useless. Moreover, since the nodes in WSNs use wireless communication technologies, the link is easy to be eavesdropped, and the data transmission needs to be encrypted. So big data transmission in WSNs should solve the energy-efficiency and encryption problems. CS can decrease the volume of the data transmitted, which can save the energy to prolong the life of the node [4, 5]. CS is also a kind of encryption method resulting from the randomness of the measurement matrix [6]. So CS can be used to encrypt data and save energy simultaneously.

However, the CS encryption uses the measurement matrix as the secret key, while the measurement matrix needs a huge storage space which is not suitable for WSNs. In WSNs, nodes are not only energy-limited, but also storage and calculation resources-constrained. Many optimization methods for the measurement matrix have been proposed [7, 8]. However, most of these existing methods focused on how to improve the recovery accuracy, decrease the iteration number, and accelerate the calculation. For reducing the size of the measurement matrix, most works focused on reducing the row number of the measurement matrix [9, 10], because the column number must be equal to the signal length according to the rule of matrix multiplication. Another kind of method to reduce the matrix size is dividing the signal into blocks, but this needs extra data processing overhead.

Another kind of CS encryption can save storage space by storing matrix generation parameters as the secret key rather than the whole matrix [11, 12]. This kind of CS encryption generates matrices by deterministic methods such

as algebraic curves [13], coding (LDPC, BCH) [14], and chaotic systems (Chebyshev, Logistic, and Tent) [15, 16]; it can save huge storage space compared with keeping the whole matrix. However, using this method, users have to generate the matrix before encryption for each transmission. Although the deterministic method can decrease the key storage space by storing parameters, it needs to calculate the measurement matrix in real-time, which is at the expense of the calculation resource.

In this paper, semitensor product compressive sensing (STP-CS) is proposed to solve the problems above. STP-CS can save the storage space by introducing the semitensor product [17–19] into compressive sensing, which can break through the dimension restriction of matrix multiplication and reduce the row and column numbers of the measurement matrix simultaneously. Compared with deterministic methods, the calculation resource of STP-CS is saved, because STP-CS does not need to generate the matrix in real-time before data encryption. An algorithm for STP-CS is also proposed, which saves the calculation resource compared with the traditional CS in theory and under simulation. Contributions of this paper are as follows:

(i) STP-CS reduces the row and column numbers of the measurement matrix simultaneously to save the storage space.

(ii) An algorithm of STP-CS is proposed to save the computing resources.

(iii) The recovery performance of STP-CS is similar to those of the traditional CS and CCS, and the compression ratio performance of STP-CS is not affected.

The rest of this paper is organized as follows. Section 2 introduces the details of STP-CS encryption. The storage and calculation resources of STP-CS are analyzed in Section 3. Simulation results are discussed in Section 4. The last section concludes this paper.

Notation. The following notation is used throughout the paper. WSN denotes the wireless sensor network. CS denotes compressive sensing. CCS denotes the chaotic compressive sensing. STP-CS denotes the semitensor product compressive sensing. x, y denote the plain message and cipher message, respectively. P, K are the length and sparsity of x, respectively. Φ denotes the measurement matrix, and M, N are the row and column number of the measurement matrix, respectively.

2. Related Works

In this section, some works about how to decrease the storage space of the CS secret key are introduced. There are two kinds of methods to decrease the storage space; one kind is reducing the size of the measurement matrix. Another is using the deterministic measurement matrix, and with this method, the matrix generation parameters are saved rather than the whole matrix.

A method for designing the measurement matrix is proposed in [10]. This method can reduce the row number of the measurement matrix, but the side information is needed for the design of the measurement matrix. Model based compressive sensing is proposed in [9]. Using this model based CS, the signal can be recovered by less number of measurements by leveraging more realistic signal models; less number of measurements means the row number of measurement matrix is reduced. But the recovery algorithm has to be improved; the traditional recovery algorithm cannot be used. Compared with these methods, STP-CS can reduce not only the row number of the measurement matrix, but also the column number by breaking through the restriction of matrix multiplication.

There are many kinds of deterministic measurement matrices. The chaotic sequence has the property of pseudo-random, so it can be used for constructing the measurement matrix [15]. The possibility of constructing measurement matrix with different kinds of chaotic systems is investigated, including Chen system, Chua system, and Lorenz system [16]. Algebraic curves like elliptic curves can also be used to construct the deterministic measurement matrices [13]. LDPC code is another kind of method for constructing the deterministic measurement matrices [14]. All these methods only need to store some parameters rather than the whole matrix, but the measurement matrix has to be generated in real-time. Compared with these methods, STP-CS can save huge computing resource.

In addition, there are many other matrices which can be used as deterministic measurement matrix, such as cyclic matrix [20], Toeplitz matrix [21], chirp matrix [22], and polynomial matrix [23]. However, these matrices have other restrictions. Cyclic matrix and Toeplitz matrix still need to store lots of test data, and the construction of polynomial matrix is limited by the signal length [20].

3. STP-CS Data Communication

In this section, the details of our proposed STP-CS encryption are introduced. Before this, CS encryption is introduced.

3.1. CS Encryption. Based on CS theory [24, 25], suppose $x \in R^N$ is a plain message; project x to $y \in R^M$ using the matrix $\Phi \in R^{M\times N}$, $y = \Phi x$, where Φ is called the measurement matrix and $M < N$. Because y is very different from x, y is regarded as the cipher message, and Φ is the secret key. At the receiver, x can be recovered with y and Φ by utilizing some algorithms such as BP, OMP, and ROMP [24, 26, 27]. For the recovery, x should be sparse or sparse on some orthogonal basis $\Psi \in R^{N\times N}$; that is, $x = \Psi s$. The sparsity here means K values of s are nonzero, while the other $N - K$ values are zero, where $K \ll N$. Though $M < N$, for the accuracy recovery, M cannot be arbitrarily small; it has to be satisfied with

$$M \geq cK \log_2\left(\frac{N}{K}\right), \tag{1}$$

where c is a small constant [24]. Resulting from the dimension restriction of the matrix multiplication, the column number N of the measurement matrix has to be equal to the dimension of the signal x. For storing a CS secret key, MN elements

need to be stored. So, to decrease the size of the measurement matrix, this restriction has to be broken through.

3.2. Semitensor Product. The semitensor product (STP) was proposed by Cheng and Zhang in [17]. STP is the generalization of the conventional matrix multiplication, and it can break through the dimension match restriction of the conventional matrix multiplication.

Suppose u is a row vector of dimension np; v is a column vector of dimension p; dividing u to p equal parts, that is, $u^1, \ldots, u^p$, each part u^i is a row vector of dimension n. The definition of STP, denoted by $\ltimes$, is

$$u \ltimes v = \sum_{i=1}^{p} u^i v_i \in R^{1\times n}. \tag{2}$$

Similarly, $v^T \ltimes u^T = \sum_{i=1}^{p} v_i (u^i)^T \in R^{n\times 1}$. Generalized to a matrix, suppose $A \in R^{m\times n}$, $B \in R^{p\times q}$; if n is the factor of p or p is the factor of n, the definition of the semitensor product of A and B is as follows:

$$A \ltimes B = \begin{bmatrix} A_1 \ltimes B^1 & \cdots & A_1 \ltimes B^q \\ \vdots & \ddots & \vdots \\ A_m \ltimes B^1 & \cdots & A_m \ltimes B^q \end{bmatrix}, \tag{3}$$

where A_i denotes the ith row of A and B^j denotes the jth column of B.

3.3. STP-CS Encryption. Now, introduce STP into CS encryption [28]. The definition of STP-CS is as follows:

$$y = A \ltimes x, \tag{4}$$

where $A \in R^{M\times N}$, $M < N$, $x \in R^P$. To decrease the size of the measurement matrix A, N should be as small as possible. For meeting the requirement of STP, we choose N with the condition $N \mid P$. According to [17],

$$y = A \ltimes x = \left(A \otimes I_{P/N}\right) x, \tag{5}$$

where $y \in R^{MP/N}$ and $\otimes$ denotes the Kronecker product [17]. When $N = P$, (5) translates to $y = Ax$, which is the traditional CS. From (5), STP-CS with the measurement matrix A is equivalent to the traditional CS with the measurement matrix $(A \otimes I_{P/N})$. The RIP, spark, and coherence of the measurement matrix are introduced in [28], A needs to meet these conditions. Based on the definition of STP-CS, for a signal $x \in R^P$, the column number of the measurement matrix A only needs to be satisfied with the condition $N \mid P$, while the traditional CS should meet the dimension match, and the column number must be equal to P. So compared with the traditional CS, the column number of the measurement matrix can be decreased. As for the row number of the measurement matrix in STP-CS, it can be also decreased which will be introduced in next section, while the row number of the measurement matrix in the traditional CS cannot break through the restriction in (1). So, although the STP-CS encryption also keeps the measurement matrix as the secret key, it can save a huge storage space by decreasing the size of the measurement matrix.

Compared with deterministic methods [15], like chaotic compressive sensing (CCS), the storage space of the measurement matrix in STP-CS is not decreased, because CCS stores matrix generation parameters such as chaotic parameter or chaos sequence initial value. But calculation resource of STP-CS is saved. The measurement matrix in CCS has to be generated in real-time, which will need much calculation resource. So STP-CS can save storage space compared with the traditional CS and save calculation resource compared with CCS. In fact, STP-CS can also save calculation resource compared with the traditional CS, which will be introduced in the next section. So STP-CS can be widely used in resource-limited scenarios like WSNs. STP-CS encryption can not only solve the security and energy-efficiency problems but also save storage and calculation resources.

3.4. An Algorithm for STP-CS. In this part, an algorithm for STP-CS is proposed, which can implement STP-CS using less calculation resource than the traditional CS.

From (3), computing $A \ltimes x$ needs to compute $A_i \ltimes x$, $i = 1, 2, \ldots, M$. To compute $A_i \ltimes x$, split x to P/N, and use each element of A_i to multiply the corresponding block of x, which means that every element of A needs to be multiplied by several numbers. As for matrix multiplication, suppose $\widetilde{C} = \widetilde{A}\widetilde{B}$; an arbitrary element $\widetilde{a}_{ij}$ of $\widetilde{A}$ needs to be multiplied by every element of the jth row of $\widetilde{B}$. Based on the above analysis, an algorithm for STP-CS using matrix multiplication is proposed.

(1) Project x to an $N \times (P/N)$ matrix as follows:

$$x_{\text{matrix}} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{P/N} \\ x_{1+P/N} & x_{2+P/N} & \cdots & x_{2P/N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1+P(N-1)/N} & x_{2+P(N-1)/N} & \cdots & x_{N\times P/N} \end{bmatrix}. \tag{6}$$

(2) Left multiply the above matrix x_{matrix} using the STP-CS measurement matrix A:

$$y_{\text{matrix}} = A x_{\text{matrix}}. \tag{7}$$

(3) Transform each row of y_{matrix} into a column vector, and construct a new column vector using these vectors. This new column vector is equal to y.

Next is the brief proof for step (3). Based on the second step of the algorithm, we have

$$y_{\text{matrix}}(ij) = \sum_{l=1}^{N} a_{il} x_{j+(l-1)P/N}. \tag{8}$$

And y can be split into M blocks with P/N elements; the jth element of the ith block of y is

$$y_i^j = \sum_{l=1}^{N} a_{il} x_{j+(l-1)P/N}. \tag{9}$$

So $y_i^j = y_{\text{matrix}}(ij)$, and y_{matrix} can be transformed to y.

The diagram of the STP-CS algorithm above is shown in Figure 1. x is the plain message and y is the cipher message. A is the secret key of STP-CS encryption.

4. Performance Analysis

In this section, the performance of STP-CS is analyzed, including storage resource, calculation resource, and compression ratio.

Based on (5), STP-CS with the measurement matrix $A \in R^{M\times N}$ is equivalent to the traditional CS with the measurement matrix $(A \otimes I_{P/N}) \in R^{(MP/N)\times P}$. According to (1), we have

$$\frac{MP}{N} \geqslant cK\log_2\left(\frac{P}{K}\right). \tag{10}$$

And then

$$M \geqslant \frac{cNK\log_2(P/K)}{P}, \tag{11}$$

where c is a small constant. In order to compress the signal, the dimension of y should be satisfied with $MP/N < P$; that is, $M < N$, so the range of the row number of A is

$$\frac{cNK\log_2(P/K)}{P} \leqslant M < N. \tag{12}$$

Because storing a measurement matrix needs to keep MN elements, the range of the storage space for one STP-CS key is

$$\frac{cN^2K\log_2(P/K)}{P} \leqslant MN < N^2. \tag{13}$$

Based on $N \mid P$, set $P = kN$, $k \in Z^+$, and k is the factor of P; (13) can be transformed to

$$\frac{cPK\log_2(P/K)}{k^2} \leqslant MN < \frac{P^2}{k^2}. \tag{14}$$

Equation (14) is the relationship between the key storage space and the dimension and sparsity of the signal x. For comparison, suppose $A' \in R^{M'\times N'}$ is the measurement matrix for the traditional CS. To encrypt the same signal, the condition $N' = P$, $M' \geq cK\log_2(P/K)$ should be satisfied, so the storage space for one traditional CS key is

$$cPK\log_2\left(\frac{P}{K}\right) \leqslant M'N' < P^2. \tag{15}$$

From (14) and (15), the low bound of one STP-CS key storage space is smaller than that of one traditional CS key storage space, when $k \neq 1$.

According to (4) and (5), y is a column vector of dimension MP/N. According to (3), y includes M column vectors with dimension P/N; that is,

$$y = \begin{bmatrix} y_1 & y_2 & \cdots & y_M \end{bmatrix}^T, \tag{16}$$

where $y_i = a_i \ltimes x = \sum_{j=1}^{N} a_{ij}x^j$, in which a_i is the ith row of A, and x^j is jth block of x. The dimension of x^j is P/N. From (2), computing each y_i needs $(P/N) \times N$ multiplications, that is, P multiplications, and $(N-1)\times(P/N)$ additions. y includes M y_i. So computing the whole y needs MP multiplications and $(N-1)MP/N$ additions. However, using the traditional CS needs the measurement matrix $A' \in R^{(MP/N)\times P}$ in order to get the same data volume of STP-CS. Computing each measurement needs P multiplications and $P-1$ additions. Computing the whole y needs MP^2/N multiplications and $MP(P-1)/N$ additions. To get the same number of measurements, the multiplication resources of the traditional CS are P/N times that of STP-CS; the addition resources of traditional CS are $(P-1)/(N-1)$ times that of STP-CS. Resulting from $N \mid P$, $P/N \geqslant 1$, the traditional CS needs more resources than STP-CS. When $P = N$, the resources needed are the same for both methods. In fact, if $P = N$, STP-CS degenerates to the traditional CS.

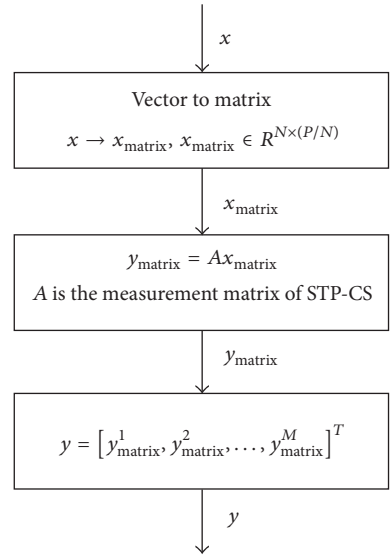


Figure 1: STP-CS algorithm diagram.

Now analyze the computing resource of the algorithm proposed in Section 2. Computing each element of y_{matrix} needs N multiplications and $N-1$ additions, and y_{matrix} has MP/N elements, so the whole resources needed for computing y_{matrix} are MP multiplications and $MP((N-1)/N)$ additions. Compared with the definition of STP-CS, the calculation quantity is the same.

Next, we analyze the compression ratio of STP-CS. From (5), the dimension of y is MP/N, so the compression ratio is $R = (MP/N)/P = M/N$, where M and N are the row number and column number of the STP-CS measurement matrix, respectively. From (12), the range of compression ratio R of STP-CS is $cK\log(P/K)/P \leqslant M/N < 1$. And the compression ratio of the traditional CS is $R' = M'/P$, where M' is the row number of the traditional CS measurement matrix, and P is the dimension of the signal. From the row number restriction, the range of R' of traditional CS is $(cK\log(P/K))/P \leqslant M'/P < 1$. So the range of STP-CS is the same as that of the traditional CS. STP-CS can obtain the same compression ratio as the traditional CS.

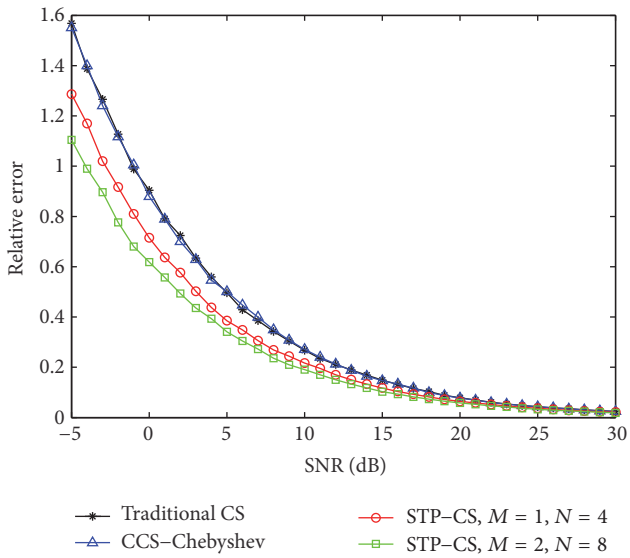


Figure 2: Recovery results. The range of SNR is from −5 dB to 30 dB. The traditional CS uses a Gaussian matrix, CCS uses a Chebyshev matrix, and STP-CS uses a Gaussian matrix. The simulation time is 200; the relative errors are the average values of the 200 simulation results.

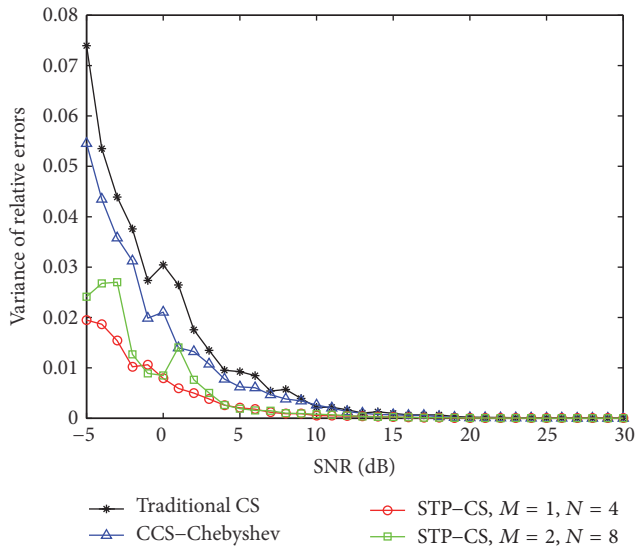


Figure 3: Variance of the relative errors of 200 simulation results. The range of SNR is from −5 dB to 30 dB. The traditional CS uses a Gaussian matrix, CCS uses a Chebyshev matrix, and STP-CS uses a Gaussian matrix.

5. Simulation Results

In this section, simulations of STP-CS encryption and decryption are discussed. In the experiment, the length of the original signal x is 256, and the sparsity K is 7. The signal is a frequency domain sparse signal, which is combined by some discrete sine signals. The recovery algorithm is OMP, and the recovery performance is measured by the relative error,

$$\delta = \frac{\|\widetilde{x} - x\|_2}{\|x\|_2}, \tag{17}$$

which is the 2-norm of the recovery error $\widetilde{x} - x$ relative to the 2-norm of the original signal x, and $\widetilde{x}$ is the recovered signal. The simulation time is 200, and the relative errors in Figures 2, 5, and 6 are the average values of the 200 simulation results.

Figure 2 shows the recovery performance of STP-CS compared with the traditional CS and CCS [16]. The compression ratio is 0.25, and two groups of the STP-CS matrix M, N are processed. The sizes of two STP-CS matrices are $M = 1$, $N = 4$ and $M = 2$, $N = 8$, respectively. The sizes of the traditional CS and CCS are both 64×256. From Figure 2, four curves coincide with each other after 20 dB. For example, At 20 dB, the relative error of the traditional CS is 0.0788, the relative error of CCS is 0.0769, the relative error of STP-CS with 1×4 matrix is 0.0644, the relative error of STP-CS with 2×8 matrix is 0.0579, and the relative errors of three kinds of matrices tend to be zero at 30 dB. Figure 3 shows the variance of the relative errors of the 200 simulation results. From Figure 3, the variance is small, after 10 dB, all variances are less than 0.01. The variance of STP-CS is smaller than the traditional CS and CCS. So the relative error of STP-CS is stable. This implies that the recovery performance of STP-CS is similar to those of the traditional CS and CCS. The size of the STP-CS matrix is extremely small, which implies that the small-size measurement matrix of STP-CS achieves a similar recovery performance as that of the traditional CS, so STP-CS can save storage space.

Not only the above signal but also STP-CS can be used for other kinds of signals. Table 1 shows the recovery results of four kinds of signals. These signals are generated by MATLAB, including Bernoulli, Gaussian, Uniform, and Power distributions. The recovery algorithm is OMP, and the measurement matrix is a 16×32 Gaussian matrix for three kinds of length signals. From Table 1, the relative errors are small for these four kinds of signals. STP-CS can encrypt the signals with different length, but the dimension of the measurement matrix of the traditional CS should be adjusted to match the signal length.

STP-CS can be also used for the image. Figure 4 shows the recovery results for a Lena image. The image size is 512×512, the size of the measurement matrix is 64×128, and the PSNR (Peak Signal to Noise Ratio) is 33.64 dB. The compression ratio of STP-CS is 0.5. For the traditional CS, the size of the measurement matrix should be 256×512 for the compression ratio 0.5, so STP-CS can save huge storage space for the measurement matrix.

The computing resources are measured by computing time. In this part, the encryption time is recorded by MATLAB system time, and the unit is millisecond. The compression ratio is also 0.25. The size of the STP-CS matrix is 1×4, the size of the traditional CS matrix is 64×256, and the size of the chaotic matrix is 64×256. The encryption time of the above three methods is 0.161 ms, 0.254 ms, and 1.953 s, respectively. Because the CCS needs to generate the matrix, the time of CCS is very long. Table 1 shows the computing time for different groups of M, N for the STP-CS matrix. From Table 2, the computing time increases with the

Table 1: Relative error of the recovery results for different signals. The encryption method is STP-CS.

Length	Signal			
	Bernoulli	Gaussian	Power	Uniform
256	2.33×10^{-16}	1.06×10^{-16}	3.10×10^{-16}	9.77×10^{-16}
512	2.42×10^{-16}	1.01×10^{-16}	3.20×10^{-16}	1.53×10^{-16}
1024	1.64×10^{-16}	2.49×10^{-16}	1.56×10^{-16}	6.31×10^{-17}

(a) (b)

Figure 4: STP-CS for image. (a) is the original image. (b) is the recovery image. The measurement matrix is a Gaussian matrix, and the recovery algorithm is OMP.

Table 2: Encryption times for different groups of M, N. The encryption method is STP-CS, unit ms.

Signal length	Matrix			
	1×4	1×8	4×16	4×32
256	0.161	0.163	0.172	0.175
512	0.163	0.163	0.171	0.172
1024	0.165	0.168	0.174	0.177
2048	0.166	0.171	0.187	0.195

increments of M, N, and P. So, to reduce the computing time, M and N should be small. Along with the increment of the signal length, the traditional CS should increase the row of the measurement matrix which will increase the calculation quantity, while the STP-CS does not need to increase the row number. So the STP-CS also has the advantage on the decrement of computing resources.

In Figure 5, the compression ratio performance of STP-CS is shown. The compression ratio of STP-CS is M/N; to get the small-size matrix, we choose M, N as small as possible. At 20 dB, the relative error of the 1×8 matrix is 0.0840, the relative error of the 1×4 matrix is 0.0655, the relative error of the 2×4 matrix is 0.0461, and the relative error of the 3×4 matrix is 0.0374. The recovery errors of these four matrices tend to be zero after 20 dB. This implies that the STP-CS can also achieve low compression ratio without affecting the recovery accuracy. Even the compression ratio is 0.125; at 30 dB, the relative error is 0.0261, similar to that of the ratio of 0.25, 0.0207. So the relative error can also tend to be zero at high SNR.

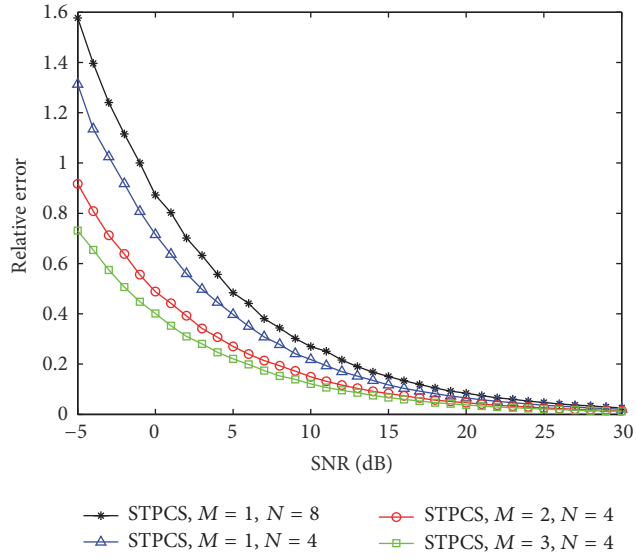


Figure 5: Performance of compression ratio. Four kinds of ratios, that is, 0.125, 0.25, 0.5, and 0.75, are tested. The simulation time is 200; the relative errors are the average values of the 200 simulation results.

From Figure 6, only the original matrix can decrypt the data correctly, and the recovery errors of the other three

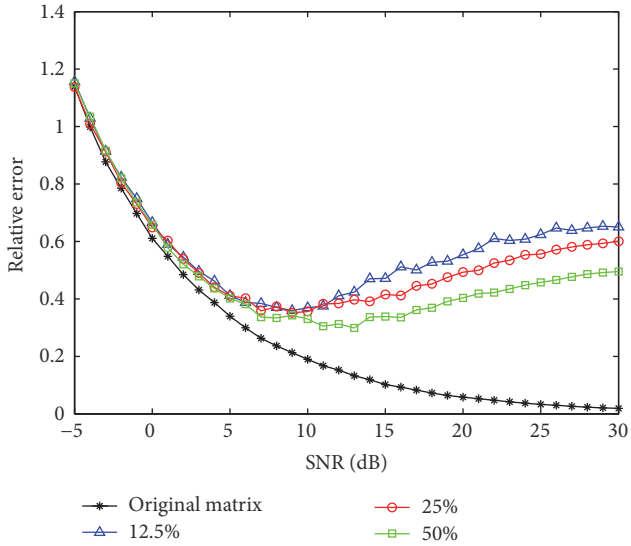


Figure 6: Security of STP-CS encryption. The data are encrypted by a 2×8 matrix. Four matrices are used to decrypt the encryption data, including the original matrix, 12.5% of the elements the same as the original matrix, 25% of the elements the same as the original matrix, and 50% of the elements the same as the original matrix. The left unknown elements are generated randomly.

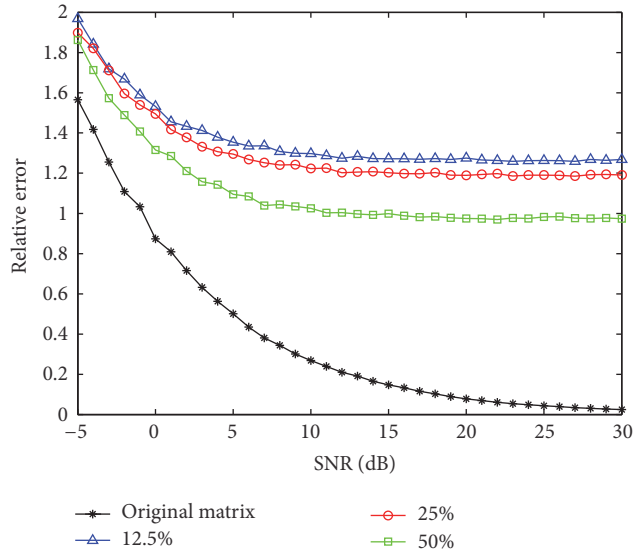


Figure 7: Security of the traditional CS encryption. The data are encrypted by a 64×256 matrix. Four matrices are used to decrypt the encryption data, including the original matrix, 12.5% of the elements the same as the original matrix, 25% of the elements the same as the original matrix, and 50% of the elements the same as the original matrix. The left unknown elements are generated randomly.

matrices are larger than 20% from −5 dB to 30 dB. The elements of other three matrices are only partly the same as those of the original matrix, and the encrypted data cannot be decrypted by a different key. Even if there is an eavesdropper who has 50% of the elements of the key, the encrypted data still cannot be decrypted. At 30 dB, the relative errors of these three matrices are larger than 40%, which implies that, even at high SNR, the eavesdropper still cannot recover the data accurately.

For comparison, Figure 7 shows the security of the traditional CS. Similar to the encryption of STP-CS, only the original matrix can decrypt the data correctly, and the other three matrices cannot decrypt the data; the recovery errors of the other three matrices are larger than 80% from −5 dB to 30 dB. Based on this relative error, the performance of the traditional CS is better than STP-CS. But the dimension of the measurement matrix is 64×256, while the dimension of the measurement matrix is 2×8, so the security performance of STP-CS can be improved by increasing the size of the measurement matrix.

6. Conclusions

CS can fulfill the energy-efficiency and the encryption for big data transmission simultaneously. But the measurement matrix needs huge storage space, and the calculation cost of CS is large. In this paper, we propose STP-CS encryption to decrease the storage space for the secret key to save storage resource and reduce the calculation amount to save calculation resource. The simulation results show that the performance of saving resource is better compared with the traditional CS and CCS.

Conflicts of Interest

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

Acknowledgments

This paper is supported by the National Key Research and Development Program of China (Grants nos. 2016YFB0800602 and 2016YFB0800604), the National Natural Science Foundation of China (Grants nos. 61573067 and 61472045), the Beijing City Board of Education Science and Technology Project (Grant no. KM201510015009), and the Beijing City Board of Education Science and Technology Key Project (Grant no. KZ201510015015).

References

[1] Y. Lee, D. Blaauw, and D. Sylvester, "Ultralow power circuit design for wireless sensor nodes for structural health monitoring," *Proceedings of the IEEE*, vol. 104, no. 8, pp. 1529–1546, 2016.

[2] I. L. Santos, L. Pirmez, L. R. Carmo et al., "A decentralized damage detection system for wireless sensor and actuator networks," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1363–1376, 2016.

[3] X. Ding, Y. Tian, and Y. Yu, "A real-time big data gathering algorithm based on indoor wireless sensor networks for risk analysis of industrial operations," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1232–1242, 2016.

[4] L. Quan, S. Xiao, X. Xue, and C. Lu, "Neighbor-aided spatial-temporal compressive data gathering in wireless sensor networks," *IEEE Communications Letters*, vol. 20, no. 3, pp. 578–581, 2016.

[5] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156–166, 2012.

[6] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[7] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Sanei, "On optimization of the measurement matrix for compressive sensing," in *Proceedings of the 18th European Signal Processing Conference*, pp. 427–431, August 2010.

[8] S. Sharma, A. Gupta, and V. Bhatia, "A new sparse signal-matched measurement matrix for compressive sensing in uwb communication," *IEEE Access*, vol. 4, pp. 5327–5342, 2016.

[9] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[10] P. Song, J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Measurement matrix design for compressive sensing with side information at the encoder," in *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP '16)*, pp. 1–5, IEEE, Palma de Mallorca, Spain, June 2016.

[11] R. R. Naidu, P. Jampana, and C. S. Sastry, "Deterministic compressed sensing matrices: construction via Euler squares and applications," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3566–3575, 2016.

[12] A. Ravelomanantsoa, H. Rabah, and A. Rouane, "Compressed sensing: a simple deterministic measurement matrix and a fast recovery algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405–3413, 2015.

[13] S. Li, F. Gao, G. Ge, and S. Zhang, "Deterministic construction of compressed sensing matrices via algebraic curves," *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5035–5041, 2012.

[14] J. Zhang, G. Han, and Y. Fang, "Deterministic construction of compressed sensing matrices from protograph LDPC codes," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1960–1964, 2015.

[15] L. Yu, J. P. Barbot, G. Zheng, and H. Sun, "Compressive sensing with chaotic sequence," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 731–734, 2010.

[16] G. Chen, D. Zhang, Q. Chen, and D. Zhou, "The characteristic of different chaotic sequences for compressive sensing," in *Proceedings of the 5th International Congress on Image and Signal Processing (CISP '12)*, pp. 1475–1479, IEEE, Chongqing, China, October 2012.

[17] D. Cheng and L. Zhang, "On semi-tensor product of matrices and its applications," *Acta Mathematicae Applicatae Sinica*, vol. 19, no. 2, pp. 219–228, 2003.

[18] D. Cheng, H. Qi, and A. Xue, "A survey on semi-tensor product of matrices," *Journal of Systems Science and Complexity*, vol. 20, no. 2, pp. 304–322, 2007.

[19] D. Cheng and Y. Dong, "Semi-tensor product of matrices and its some applications to physics," *Methods and Applications of Analysis*, vol. 10, no. 4, pp. 565–588, 2003.

[20] H. Yuan, H. Song, X. Sun, K. Guo, and Z. Ju, "Compressive sensing measurement matrix construction based on improved size compatible array LDPC code," *IET Image Processing*, vol. 9, no. 11, pp. 993–1001, 2015.

[21] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing (SSP '07)*, pp. 294–298, IEEE, Madison, Wis, USA, August 2007.

[22] L. Applebaum, S. D. Howard, S. Searle, and R. Calderbank, "Chirp sensing codes: deterministic compressed sensing measurements for fast recovery," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, 2009.

[23] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[24] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[25] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[26] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers (ASILOMAR '08)*, pp. 581–587, IEEE, Pacific Grove, Calif, USA, October 2008.

[27] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[28] D. Xie, H. Peng, L. Li, and Y. Yang, "Semi-tensor compressed sensing," *Digital Signal Processing*, vol. 58, pp. 85–92, 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1975719, 14 pages
https://doi.org/10.1155/2017/1975719

Research Article

New Collaborative Filtering Algorithms Based on SVD++ and Differential Privacy

Zhengzheng Xian,[1,2] Qiliang Li,[1] Gai Li,[3] and Lei Li[1]

[1]*School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, China*
[2]*Guangdong University of Finance, Guangzhou, Guangdong, China*
[3]*Shunde Polytechnic, Foshan, Guangdong, China*

Correspondence should be addressed to Zhengzheng Xian; xianzhengzheng@126.com

Received 28 November 2016; Revised 5 February 2017; Accepted 19 February 2017; Published 19 March 2017

Academic Editor: Kaoru Ota

Collaborative filtering technology has been widely used in the recommender system, and its implementation is supported by the large amount of real and reliable user data from the big-data era. However, with the increase of the users' information-security awareness, these data are reduced or the quality of the data becomes worse. Singular Value Decomposition (SVD) is one of the common matrix factorization methods used in collaborative filtering, which introduces the bias information of users and items and is realized by using algebraic feature extraction. The derivative model SVD++ of SVD achieves better predictive accuracy due to the addition of implicit feedback information. Differential privacy is defined very strictly and can be proved, which has become an effective measure to solve the problem of attackers indirectly deducing the personal privacy information by using background knowledge. In this paper, differential privacy is applied to the SVD++ model through three approaches: gradient perturbation, objective-function perturbation, and output perturbation. Through theoretical derivation and experimental verification, the new algorithms proposed can better protect the privacy of the original data on the basis of ensuring the predictive accuracy. In addition, an effective scheme is given that can measure the privacy protection strength and predictive accuracy, and a reasonable range for selection of the differential privacy parameter is provided.

1. Introduction

The Internet has been widely used since the birth of Web 2.0, and the human lifestyle has been greatly changed. When a user opens a shopping website or a mobile terminal application, a very enthusiastic recommender system will list some commodities in which he or she may be interested based on the purchase history record, browser footprint, evaluation information, and so forth. Today, there are numerous intelligent applications such as those. If the value of implicit feedback information such as historical browsing data, historical rating data, and the evaluation timestamp can be fully exploited, the predictive accuracy could be improved further. The Singular Value Decomposition (SVD) model [1] is a kind of common collaborative filtering method to provide personalized recommendation services, and the predictive accuracy can be improved by considering the user and item bias information. As a derivative model of SVD, the SVD++ model [2–4] achieves better recommendation accuracy by adding implicit feedback information, such as movies that a user has evaluated, and the specific value of the score does not matter for this kind of information.

While the Internet has brought much convenience to users, their daily medical, transportation, purchase, and Internet browsing information, which is neglected by the users themselves, will all be recorded to become data resources for Internet companies to identify further business opportunities and benefits. Meanwhile, there is also a risk of leakage of personal privacy information because the information is collected. In recent years, the issue of leakage of personal privacy information triggered by the Internet has arisen frequently. For example, in the Netflix Prize competition, the Netflix Corporation released a dataset through anonymous processing. However, researchers from

the University of Texas were able to deduce the real Netflix users by linking the rating and timestamp in this dataset with public information on Internet Movie Database (IMDB). As another example, in 2012, an American college student was recognized as homosexual by his roommate. His roommate used a network to search for the frequency of access to homosexual forums and websites. Collaborative filtering based on items that are related in a transaction performed by a user will lead to the increase in similarity with this user's previous commodity transactions. Thus, an attacker can track similar commodity lists related to the target user (attack target) and then determine what is a new commodity. When a similar commodity appears in these lists, the attacker can deduce the item to be added to the target user's records. Thus, what can be obtained through indirect derivation of the personal privacy information is increasingly considered.

In 2006, Dwork [5] proposed differential privacy (DP), and it can solve the issues of leakage of personal privacy information by relating to the background knowledge mentioned above. It has a very strict definition and has nothing to do with background knowledge, so it can fundamentally solve the defects of the traditional privacy protection model and is an effective way to remove the possibility of leakage of personal privacy information from the data source. Although DP has been researched for 10 years, the major research achievements are academic theories. The Apple corporation has always claimed that the user's privacy should be the top priority. This year, at the Worldwide Developers Conference (WWDC2016), Apple proposed the application of DP to collect and analyse user data from the keyboard, Spotlight, and Notes in iOS 10. Its goal is to ensure that the Quality of Service (QoS) [6] will not be affected and that the user's personal information will not be leaked. This measure opens up new pioneering work on DP in the application layer.

Today, it is quite urgent in the field of data mining to improve QoS and ensure the security of personal privacy information, eliminating users' worries and providing true and reliable data in order to guarantee the production of effective knowledge and rules [7, 8].

The contributions of our work are summarized as follows. First, we propose three new methods that apply differential privacy to SVD++ through gradient perturbation, objective-function perturbation, and output perturbation. Second, rigorous mathematical proofs are given to ensure that they all maintain the differential privacy. Third, we compare the predictive accuracies obtained by our differential privacy algorithms for SVD++ with those of the same methods for SVD and related methods in the literature on two real datasets and the method of objective perturbation for SVD++. Results show that our methods obtain better results in terms of balancing privacy and prediction. Finally, we propose a scheme for selection of DP protection parameter ε in order to balance the strength of privacy and the predictive accuracy, and a reasonable range of DP parameter ε could be obtained by this scheme.

The remainder to the paper is organized as follows. Section 2 surveys some works related to private-preserving in recommender systems. Section 3 introduces the SVD++ model and DP model. Section 4 presents the three new methods, which apply DP to SVD++ using gradient perturbation, objective-function perturbation, and output perturbation. Section 5 presents the experimental evaluation of each method on two real datasets. Finally, Section 6 summarizes the key aspects of our work and briefly addresses the directions for future work.

2. Related Work

The privacy protection of recommender systems became a popular research topic when Canny [11] proposed that the recommender not use the user's data for financial benefit in 2002. It is a hot topic in research to apply DP to personalized collaborative filtering technology since DP is considered to be the best privacy protection technology. McSherry and Mironov [12] applied DP to collaborative filtering first, and the main idea of the paper was to use the Laplace mechanism to compute a differential private item-to-item covariance matrix, which was used to find neighbours and compute the SVD recommendation. However, it seems unreasonable that there is less contribution to the covariance when a user's buying activity increases. Zhu et al. [13] addressed the privacy issues in the context of neighbourhood based CF methods by proposing a Private Neighbour Collaborative Filtering (PNCF) algorithm. Hua et al. [14] first proposed that recommenders who are not trusted should be prevented from using a user's ratings, while allowing the user to leave or join in the matrix factorization (MF) process and then realizing DP protection by disturbing the objective function of MF. Liu et al. [15] proposed a method that applied DP to Bayesian posterior sampling by Stochastic Gradient Langevin Dynamics (SGLD), thus avoiding the influence of the Gaussian noise on the whole parameter space. Zhu and Sun [16] proposed Differentially Private Item-Based Recommendation and Differentially Private User-Based Recommendation and designed a low-sensitivity metric to measure the similarities between both items and users. Yan et al. [17] proposed a socially aware algorithm called DynaEgo to improve the performance of privacy-preserving collaborative filtering. DynaEgo utilizes the principle of DP as well as the social relationships to adaptively modify the users' rating histories to prevent exact user information from being leaked. Javidbakht and Venkitasubramaniam [18] proposed using DP as a metric to quantify the privacy of the intended destination, and optimal probabilistic routing schemes are investigated under unicast and multicast paradigms. Balu and Furon [19] proposed using sketching techniques to implicitly provide DP guarantees by taking advantage of the inherent randomness of the data structure, and this approach is well suited for large-scale applications. Berlioz et al. [9] applied DP to the latent factor model for each step of MF; however, they did not provide rigorous mathematical proofs and need to do some preprocessing of the raw data; thus, the experimental results showed that a large DP parameter is needed to obtain good predictive accuracy.

Chaudhuri et al. [20] proposed general techniques to produce privacy-preserving approximations of classifiers learned via (regularized) Empirical Risk Minimization (ERM). They

proposed an output perturbation and objective-function perturbation based DP model but these methods were applied to logistic regression and SVM in [20]. Based on the above works, the SVD++ model, which is a derivative model of SVD, is the research object, and three new algorithms that apply DP to SVD++ using gradient perturbation, objective-function perturbation, and output perturbation are proposed. To improve the predictive accuracy, SVD++ considers the related information of the user and item. The theoretical proofs are given and the experiment results show that the new private SVD++ algorithms obtain better predictive accuracy, compared with the same DP treatment of traditional MF [9] and SVD.

The DP parameter is the key to the privacy protection power, but in the current study, it was selected by experience. Finally, an effective trade-off scheme is given that can balance the privacy protection and the predictive accuracy to a certain extent and can provide a reasonable range for parameter selection.

3. Preliminaries

3.1. SVD++ Model. The "user-item" rating matrix is the core data used by the recommender system. MF is a good method of predicting the missing ratings in collaborative filtering. In brief, MF involves factorizing a sparse matrix and finding two latent factor matrices: the first is the user matrix to indicate the user's features (i.e., the degree of preference of a user for each factor) and the other is the item matrix, which indicates the item's features (i.e., the weight of an item for each factor). The missing ratings are then predicted from the inner product of these two factor matrices.

Let $R_{n\times m}$ be a rating matrix containing the ratings of n users for m items. Each matrix element r_{ui} refers to the rating of user u for item i. Given a lower dimension d, MF factorizes the raw matrix $R_{n\times m}$ into two latent factor matrices: one is the user-factor matrix $P_{n\times d}$ and the other is the item-factor matrix $Q_{d\times m}$. The factorization is done such that R is approximated as the inner product of P and Q (i.e., $\widetilde{R}_{n\times m} = P_{n\times d}\times Q_{d\times m}$), and each observed rating r_{ui} is approximated by $\widetilde{r}_{ui} = q_i^T \cdot p_u$ (also called the predicted value). However, $q_i^T \cdot p_u$ only captures the relationship between the user u and the item i. In the real world, the observed rating may be affected by the preference of the user or the characteristics of the item. In other words, the relationship between the user u and the item i can be replaced by the bias information. For instance, suppose one wants to predict the rating of the movie "Batman" by the user "Tom." Now, the average rating of all movies on one website is 3.5, and Tom tends to give a rating that is 0.3 lower than the average because he is a critical man. The movie "Batman" is better than the average movie, so it tends to be rated 0.2 above the average. Therefore, considering the user and movie bias information, by performing the calculation $3.5 - 0.3 + 0.2 = 3.4$, it is predicted that Tom will give the movie "Batman" a rating of 3.4. The user and item bias information can reflect the truth of the rating more objectively. SVD is a typical factorization technology (known as a baseline predictor in some works in the literature). Thus, the predicted rating is changed to

$$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot p_u, \tag{1}$$

where μ is the overall average rating and b_u and b_i indicate the observed deviations of user u and item i, respectively.

The goal of a recommender system is to improve the predictive accuracy. In fact, the user will leave some implicit feedback information, such as historical browsing data, and historical rating data, on Web applications as long as any user has rated item i, no matter what the specific rating value is. To a certain extent, the rating operation already reflects the degree of a user's preference for each latent factor. Therefore, the SVD++ model introduces the implicit feedback information based on SVD; that is, it adds a factor vector ($y_j \in R^f$) for each item, and these item factors are used to describe the characteristics of the item, regardless of whether it has been evaluated. Then, the user's factor matrix is modelled, so that a better user bias can be obtained. Thus, the predictive rating of the SVD++ model is

$$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot \left(p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j \right), \tag{2}$$

where $R(u)$ is the number of items rated by user u.

To obtain the optimal P and Q, the regularized squared error can be minimized as follows. The objective function of the SVD++ model is

$$\min_{P,Q} \sum_{r_{ui}\in R} \left[r_{ui} - \mu - b_u - b_i - q_i^T \cdot \left(p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j \right)^2 + \lambda \left(b_u^2 + b_i^2 + \|p_u\|^2 + \|q_i\|^2 \right) \right], \tag{3}$$

where λ is the regularization parameter to regularize the factors and prevent overfitting.

With regard to b_u, b_i, and $\sum y_j$, two methods can be used [1]: fast empirical likelihood estimation (i.e., formula (4)) and Stochastic Gradient Descent (SGD). Considering the rate of convergence and the influence of the error in each iteration, the first method is used in this paper.

$$b_i = \frac{\sum_{u\in R(i)} (r_{ui} - \mu)}{\lambda_1 + |R(i)|},$$

$$b_u = \frac{\sum_{i\in R(u)} (r_{ui} - \mu - b_i)}{\lambda_2 + |R(u)|} \tag{4}$$

$$\sum_{j\in R(u)} y_j = \frac{\sum_{j\in R(u)} I(r_{uj} > 0)}{\lambda_3 + |R(u)|}.$$

In formula (4), when $r_{uj} > 0$, the value of $I(r_{uj} > 0)$ will be 1; otherwise, it will be 0. In addition, averages tend to zero using the regularization parameters λ_1, λ_2, and λ_3, which are determined by cross-validation.

SGD and Alternating Least Squares (ALS) are two common optimization algorithms used to solve the objective function (formula (4)). The SGD algorithm is a combination of randomness and optimization and does not need to calculate the exact value but uses unbiased estimation.

Stochastic Gradient Descent. Let e_{ui} represent the error between the true and the predicted values (i.e., $e_{ui} = r_{ui} - \tilde{r}_{ui}$). p_u is any element of the user matrix P, q_i is any element of the item matrix Q, and the error of SVD++ can be expressed as $e_{ui} = r_{ui} - (\mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum_{j \in R(u)} y_j))$. In SGD, the factors are learned by iteratively evaluating the error e_{ui} for each rating r_{ui}, and the user and item vectors are updated by taking a step in the direction opposite to the gradient of the regularized loss function. Then, the updating rules for both p_u and q_i can be formulated as follows:

$$\begin{aligned} p_u &\longleftarrow p_u + \gamma (e_{ui} q_i - \lambda p_u), \\ q_i &\longleftarrow q_i + \gamma (e_{ui} p_u - \lambda q_i), \end{aligned} \tag{5}$$

where constant γ is the learning rate and can determine the rate of error minimization.

Alternating Least Squares. In ALS, the optimization problem can be solved iteratively. One latent matrix (say P) in each iteration is fixed and then the objective function of SVD++ (formula (3)) is converted into a convex optimization problem, where the solution (say Q) can be found efficiently. Similarly, another latent matrix can be found in the same way. Finally, these steps are repeated until convergence is achieved.

3.2. Differential Privacy. The privacy protection of the collaborative filtering algorithm needs not only to reduce the risk of leaking the private information from the original data but also to ensure the availability of data. DP defines an extremely strict attack model and provides a rigorous, quantitative representation and proof of the risk of leakage of private information. The amount of background knowledge that the attacker has does not matter since DP protects information of the user's potential privacy by adding noise in order to prevent the attacker from inferring the user's protected information even if the attacker knows other information. The attacker does not know whether certain user information exists in the original dataset. Because DP can result in recommendation results not related to the information in the original dataset, DP is applied to the recommender system based on collaborative filtering to prevent indirect deduction of personal private information.

Definition 1 (ε-differential privacy). Given any two adjacent "user-item" rating matrices $R_{n\times m}$ and $R'_{n\times m}$, which differ by at most one score, if any possible output result S ($S \in \text{Range}(A)$) satisfies formula (6), the random algorithm A provides ε-differential privacy.

$$\Pr\left[A\left(R_{n\times m}\right) \in S\right] \leq \exp(\varepsilon) \times \Pr\left[A\left(R'_{n\times m}\right) \in S\right], \tag{6}$$

where $\Pr[\cdot]$ is the probability that private information will be disclosed and is controlled by the randomness of algorithm A; it is independent of the background knowledge of the attacker. Parameter ε is used to indicate the strength of privacy protection, where a smaller value indicates a higher strength of privacy protection. In addition, the two rating matrices differ by at most one score and can also be understood as two matrices that differ by at most one record of a user.

The key technology of DP protection is to add noise that satisfies the Laplace or exponent mechanism [21]. The former is applied to the results for numerical protection and the latter is applied for nonnumerical protection. The amount of noise is related to the function's sensitivity and the privacy protection parameter ε. The sensitivity of the function is that the maximum difference in the output results comes from two datasets that differ by only one record. The sensitivity is divided into global sensitivity and local sensitivity. The former is determined by the function itself and different functions will have different global sensitivities. The latter is determined by the specific given dataset and the function itself. The formal definition of global sensitivity, the Laplace mechanism, and the two composition properties of DP are given as follows.

Definition 2 (global sensitivity). Given any two adjacent "user-item" rating matrices $R_{n\times m}$ and $R'_{n\times m}$ that differ by at most one score, for any function $f : (R_{n\times m}, I) \to \mathbb{R}$, the L_k-global sensitivity of function f is

$$\text{GS}_f = \max_{R,R'} \left\| f(R, i) - f\left(R', i\right) \right\|_k, \tag{7}$$

where d is the dimension of function f, $f(R, i)$ is the predicted value of item i, and $\| \cdot \|_k$ denotes the L_k-norm.

If the global sensitivity of the function is too large to compute the average, median, and so forth, enough noise must be added to protect the privacy, but this will lead to the reduction in the availability of data. To address this problem, Nissim et al. [22] proposed the local sensitivity. In this paper, global sensitivity is adopted because the sensitivity of our function is small.

Dwork et al. [21] demonstrated that the Laplace mechanism could be used to obtain ε-differential privacy. The main idea is to add noise sampled from a Laplace distribution with a calibrated scale b. The probability density function of the Laplace distribution with mean 0 and scale b is

$$f(x \mid b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{8}$$

In this paper, it is denoted as lap(b).

Theorem 3. *Given any two adjacent "user-item" rating matrices $R_{n\times m}$ and $R'_{n\times m}$ that differ by at most one score, for any function $f : (R_{n\times m}, I) \to \mathbb{R}$ (its global sensitivity is GS_f), if the random noise $Y \sim Lap(\mathrm{GS}_f/\varepsilon)$, and the algorithm A satisfy*

$$A(R, i) = f(R, i) + Y, \tag{9}$$

the algorithm A provides ε-differential privacy.

This work also relies on the K-norm mechanism [23], which makes it possible to calibrate noise to the L_2-sensitivity of the evaluated function.

In this paper, the outputs of the new privacy algorithms are all numerical, so the Laplace mechanism is used to achieve DP.

Composition. Usually, a complex privacy-preserving problem requires DP protection technology to be applied multiple times. In this case, in order to ensure that the privacy protection level of the whole process is controlled within the budget given by the privacy protection parameter ε, two important composition properties of DP itself are required. One is the sequential composition property, and the other is the parallel composition property [21]. The sequential composition property ensures that multiple random algorithms are distributed in a DP budget (like $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$), and each algorithm maintains ε_i-differential privacy. For the same dataset, the composition algorithm of these algorithms will maintain the sum of the total privacy budget DP (i.e., it will maintain $(\sum_i \varepsilon_i)$-differential privacy). The parallel composition property means that, for a disjoint dataset, the composition algorithm of these algorithms will maintain the maximum total privacy budget DP (i.e., it will maintain $(\max \varepsilon_i)$-differential privacy).

4. Privacy-Preserving SVD++

The intuitive idea is that, after using traditional MF to solve this problem, there should be some latent features that determine how a user rates an item. However, if an attacker has some background knowledge, he or she can obtain the user's private data from the original rating matrix. For example, an attacker can infer that a user likes certain types of movies, but the user does not want other people to know this. Thus, our goal is to protect the raw rating matrix by using DP reasonably. The main idea of SVD++ is to analyse the user's preference for each factor and the extent to which the film contains the various factors from the observed ratings and some implicit feedback from users and then to predict the missing score. In this paper, considering the fact that SVD can obtain good predictive accuracy, we apply DP to SVD++ flexibly. Similarly, to the traditional MF, the SVD++ process can also be divided into the following four stages:

(i) Inputting of the original rating matrix

(ii) SVD++ factorization process by SGD or ALS

(iii) Outputting of the user characteristic matrix and the item characteristic matrix

(iv) Rating prediction (i.e., recommendation)

In [9, 10], DP was applied to these four stages and it was necessary to perform some preprocessing of the original matrix. The work of [10] was an extension of [9], and several algorithms in these two works are the same. Compared with [9, 10], our algorithms have three advantages. The first is that our algorithms do not perform any preprocessing with DP in order to ensure the availability of the original data. The second is that our algorithms adopt SVD++ to achieve MF because the SVD++ model considers the user and item biases and implicit feedback information of users in order to improve the recommendation accuracy. The third is that the objective perturbation of ALS for SVD++ comes from the idea of [20] and obtains better experimental results on two datasets than [9, 10].

4.1. SGD with Gradient Perturbation for SVD++. SGD with gradient perturbation for SVD++ applies DP to the error of each iteration in the SGD optimization algorithm. For a detailed description of the process, see Algorithm 1.

For Algorithm 1, a few explanatory points need to be stated as follows:

(1) To constrain the effect of noise, the obtained error can be to a range (in our experiments, we let $e_{\max} = 2$ and $e_{\min} = -2$ due to the experimental rating being between 1 and 5).

(2) The number of gradient descent iterations k should be given in advance.

(3) According to the sequential composition property of DP, the noise at each iteration is calibrated to maintain (ε/k)-differential privacy so that the overall SVD++ maintains ε-differential privacy after k iterations.

Theorem 4. *Given the differential privacy parameter ε and the maximum value ($r_{\max}$) and minimum value ($r_{\min}$) in the "user-item" rating matrix, set $\Delta = r_{\max} - r_{\min}$ and let the rating error in each iteration be $e_{ui} = r_{ui} - \tilde{r}_{ui}$ (r_{ui} is the raw rating and $\tilde{r}_{ui}$ is the predictive rating). If the noise vector is $v(b) \propto \exp(-\varepsilon\|b\|/(\Delta k))$, then Algorithm 1 provides ε-differential privacy after k iterations.*

Proof. First, the error ($e_{ui} = r_{ui} - \tilde{r}_{ui}$) and the global sensitivity of the error ($\mathrm{GS}_{e_{ui}}$) have the largest difference between ratings, so $\mathrm{GS}_{e_{ui}} = r_{\max} - r_{\min}$.

Second, in k iterations, if the differential privacy is ε, then the budget allocated at each iteration should be ε/k.

Third, b is a noise vector that is added to e_{ui} in each iteration and its probability density is $v(b) \propto \exp(-\varepsilon\|b\|/(\Delta k))$. According to the Laplace mechanism, the new error becomes $e'_{ui} = e_{ui} + \mathrm{Lap}(\mathrm{GS}_{e_{ui}}/(\varepsilon/k)) = e_{ui} + \mathrm{Lap}(\Delta k/\varepsilon)$. Therefore, the error in each iteration maintains (ε/k)-differential privacy.

Finally, according to the sequential composition property of DP, Algorithm 1 provides $((\varepsilon/k) * k)$-differential privacy (i.e., it provides ε-differential privacy) after k iterations. □

4.2. Private-Preserving ALS for SVD++. Two new approaches were proposed in [20], namely, objective perturbation and

Input: $R_{n\times m} = \{r_{ui}\}$ – "user-item" rating matrix
d – number of factors
γ – learning rate
λ – regularization parameter of SVD++ objective function
λ_1, λ_2 and λ_3 – regularization parameters for computing the item bias, user bias, and implicit feedback factor
k – number of gradient descent iterations
$e_{\max}$ and $e_{\min}$ – upper and lower bounds on the per-rating error
ε – differential privacy parameter
Output: Latent factor matrices $P_{n\times d}$ and $Q_{d\times m}$
(1) Initialize the random latent factor matrices P and Q
(2) **for** k iterations **do**
(3) **for** each r_{ui} **do**
$b_i = \dfrac{\sum_{u\in R(i)}(r_{ui}-\mu)}{\lambda_1 + |R(i)|}$, $b_u = \dfrac{\sum_{i\in R(u)}(r_{ui}-\mu-b_i)}{\lambda_2 + |R(u)|}$
(4) $\sum\limits_{j\in R(u)} y_j = \dfrac{\sum_{j\in R(u)} I(r_{uj}>0)}{\lambda_3 + |R(u)|}$
(5) $\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum\limits_{j\in R(u)} y_i)$
(6) $e_{ui} = r_{ui} - \widetilde{r}_{ui}$
(7) $e'_{ui} = e_{ui} + b$
(where $\nu(b) \propto \exp(-\varepsilon\|b\|/\Delta k)$ and $\Delta = r_{\max} - r_{\min}$)
(8) Clamp e'_{ui} to $[e_{\min}, e_{\max}]$
(9) update $p_u : p_u \leftarrow p_u + \gamma(e'_{ui} q_i - \lambda p_u)$
(10) update $q_i : q_i \leftarrow q_i + \gamma(e'_{ui} p_u - \lambda q_i)$
(11) **end for**
(12) **end for**
(13) **return** $P_{n\times d}$ and $Q_{d\times m}$

Algorithm 1: SGD with gradient perturbation for SVD++ (DPSS++).

output perturbation using DP for the design of privacy-preserving algorithms, and then they were applied to logistic regression and SVM. Specifically, experimental results showed that the results of objective perturbation are optimal when balancing privacy protection and predictive accuracy. In this subsection, this approach is applied to the ALS optimization algorithm of SVD++. Algorithm 2 describes the process of ALS objective perturbation and Algorithm 3 describes the process of ALS output perturbation.

In the SVD++ model, considering the user's bias, the item's bias, and the rating information to which the user has contributed in which the user has taken part, then the predicted rating is changed to

$$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot \left(p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j \right) \tag{10}$$

(see Section 3.1). The basic principle of ALS for solving SVD++ can be seen in Section 3.1. According to the principle of ALS, the raw objective function (formula (3)) becomes two convex optimization problems as follows:

$$\begin{aligned} J_Q(p_u, R) &= \sum_{R_u} (r_{ui} - \widetilde{r}_{ui})^2 + n_u \lambda \|p_u\|_2^2, \\ J_p(q_i, R) &= \sum_{R_u} (r_{ui} - \widetilde{r}_{ui})^2 + n_i \lambda \|q_i\|_2^2, \end{aligned} \tag{11}$$

where R_u and R_i are subsets of raw R and

$$\begin{aligned} R_u &= \{r_{vi} \in R \mid v = u\}, \\ n_u &= |R_u|, \\ R_i &= \{r_{uv} \in R \mid v = i\}, \\ n_i &= |R_i|. \end{aligned} \tag{12}$$

Then, the main idea of Algorithm 2 is to add noise to the objective function; that is,

$$\begin{aligned} J_Q^{\text{priv}}(p_u, R) &= J_Q(p_u, R) + \frac{1}{n} b^T p_u, \\ J_P^{\text{priv}}(q_i, R) &= J_P(q_i, R) + \frac{1}{n} b^T q_i, \end{aligned} \tag{13}$$

where b is a noise vector with d components and d is the number of features of P or Q. To solve the convex optimization problem, the idea of ERM [20] is used. So, from formula (13), we can obtain

$$p_u^{\text{priv}} = \arg\min_{p_u} J_Q^{\text{priv}}(p_u, R) + \frac{1}{2}\Delta \|p_u\|^2, \tag{14}$$

$$q_i^{\text{priv}} = \arg\min_{q_i} J_P^{\text{priv}}(q_i, R) + \frac{1}{2}\Delta \|q_i\|^2. \tag{15}$$

According to Algorithm 2 of [20], the regularization terms $(1/2)\Delta\|p_u\|^2$ and $(1/2)\Delta\|q_i\|^2$ avoid overfitting after

Input: $R_{n\times m} = \{r_{ui}\}$ – "user-item" rating matrix
d – number of factors
N – total number of ratings
λ – regularization parameter of SVD++ objective function
λ_1, λ_2 and λ_3 – regularization parameters for computing the item bias, user bias, and implicit feedback factor
k – number of gradient descent iterations
ε – differential privacy parameter
C – the parameter for computing the slack term
Output: Latent factor matrices $P_{n\times d}$ and $Q_{d\times m}$
(1) Initialize random latent factor matrices P and Q:
(2) **for** k iterations **do**
(3) **for** each r_{ui} **do**

$$b_i = \frac{\sum_{u\in R(i)}(r_{ui}-\mu)}{\lambda_1 + |R(i)|},\ b_u = \frac{\sum_{i\in R(u)}(r_{ui}-\mu-b_i)}{\lambda_2 + |R(u)|}$$

(4) $$\sum_{j\in R(u)} y_j = \frac{\sum_{j\in R(u)} I(r_{uj} > 0)}{\lambda_3 + |R(u)|}$$

(5) $$\widetilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j)$$

(6) **for** each user u, when given matirx Q, **do**
(7) let $\varepsilon' = \varepsilon - \log(1 + 2C/N\lambda + C^2/N^2\lambda^2)$
(8) if $\varepsilon' > 0$ then $\Delta = 0$
(9) else $\Delta = C/N(e^{\varepsilon/4} - 1) - \lambda$, and $\varepsilon' = \varepsilon/2$
(10) Generate random noise vector b with pdf

$$\nu(b) \propto \exp\left(-\frac{\varepsilon' \|b\|}{2}\right)$$

(11) Compute $p_u^{\text{priv}} = \arg\min_{p_u} J_Q^{\text{priv}}(p_u, R) + (1/2)\Delta\|p_u\|^2$
(12) **end for**
(13) **for** each item i, when given matrix P **do**
(14) Omit (the same as (7)~(10))
(15) Compute $q_i^{\text{priv}} = \arg\min_{q_i} J_P^{\text{priv}}(q_i, R) + (1/2)\Delta\|q_i\|^2$
(16) **end for**
(17) **end for**
(18) **end for**
(19) **return** $P_{n\times d}$ and $Q_{d\times m}$

Algorithm 2: ALS with objective perturbation for SVD++ (DPSAObj++).

perturbation, where Δ is determined by the privacy parameter ε and the slack term parameter C.

The ALS objective functions for SVD++ are convex and differentiable, so they satisfy the application conditions of Algorithm 2 of [20]. In this paper, our Algorithm 2 describes the DP protection process of ALS objective perturbation to solve for the latent factors of SVD++.

Regarding Algorithm 2, a few explanatory points should be stated as follows:

(1) First, to deduce and compute the value of parameter C in steps (7) and (9), the value of C is set to 2. The specific deduction process is similar to the deduction applied in logistic regression (Corollary 4) and SVM (Corollary 6) from [20].

(2) To solve for the values of p_u and q_i after objective perturbation, that is, to solve for the partial derivatives of formulas (14) and (15), respectively, where n indicates the number of users and m indicates the number of items in the raw matrix, the key steps are as follows.

When $\forall 1 \le u \le n$ and $1 \le k \le d$, we can obtain

$$\frac{1}{2}\frac{\partial p_u^{\text{priv}}}{\partial p_{uk}} = \sum_i \Bigg(\mu + b_u + b_i + q_i^T \left(p_u + |R_u|^{-1/2} \sum_{j\in R(u)} y_j \right) - r_{ui} \Bigg) q_{ik} + \lambda n_u p_{uk} + \frac{1}{N} b_k + \frac{1}{2}\Delta p_{uk}. \tag{16}$$

Then, we have

$$\frac{1}{2}\frac{\partial p_u^{\text{priv}}}{\partial p_{uk}} = \frac{1}{2}\left(\frac{\partial p_u^{\text{priv}}}{\partial p_{u1}}, \ldots, \frac{\partial p_u^{\text{priv}}}{\partial p_{ud}}\right) = p_u \left[Q^T Q + \left(\lambda n_u + \frac{1}{2}\Delta\right) I \right]$$

Input: $R_{n\times m} = \{r_{ui}\}$ – "user-item" rating matrix
d – number of factors
λ – regularization parameter of SVD++ objective function
λ_1, λ_2 and λ_3 – regularization parameters for computing the item bias, user bias, and implicit feedback factor
k – number of gradient descent iterations
ε – differential privacy parameter
Output: Latent factor matrices $P_{n\times d}$ and $Q_{d\times m}$
(1) Initialize random latent factor matrices P and Q:
(2) **for** k iterations **do**
(3) **for** each r_{ui} **do**

$$b_i = \frac{\sum_{u\in R(i)}(r_{ui}-\mu)}{\lambda_1 + |R(i)|},\ b_u = \frac{\sum_{i\in R(u)}(r_{ui}-\mu-b_i)}{\lambda_2 + |R(u)|}$$

(4) $$\sum_{j\in R(u)} y_j = \frac{\sum_{j\in R(u)} I(r_{uj} > 0)}{\lambda_3 + |R(u)|}$$

(5) $$\tilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2} \sum_{j\in R(u)} y_j)$$

(6) **for** each user u, when given matrix Q, **do**
(7) Generate random noise vector b with pdf
(8) $$f(b) \propto \exp\left(-\frac{\varepsilon \|b\|}{2k} \cdot \frac{n_u \lambda}{2q_{\max}\Delta r}\right)$$
(9) $$p_u(R, Q) \longleftarrow \arg\min_{p_u} J_Q(p_u, R) + b$$
(10) **end for**
(11) **for** each item i, when given matrix P **do**
(12) Generate random noise vector b with pdf
(13) $$f(b) \propto \exp(-\frac{\varepsilon \|b\|}{2k} \cdot \frac{n_i \lambda}{2p_{\max}\Delta r})$$
(14) $$q_i(R, P) \longleftarrow \arg\min_{q_i} J_P(q_i, R) + b$$
(15) **end for**
(16) **end for**
(17) **end for**
(18) **return** $P_{n\times d}$ and $Q_{d\times m}$

Algorithm 3: ALS with output perturbation of SVD++ (DPSASOut++).

$$+\left(|R_u|^{-1/2} \sum_{j\in R(u)} y_j\right) Q^T Q - \left(R_u - \mu - b_u - b_i\right) Q + \frac{1}{N}\mathbf{b}, \tag{17}$$

where $n_u = |R_u|$, $R_u = \{r_{vi} \in R \mid v = u\}$, and I is a $d \times d$ identity matrix.

Then, fixing Q and solving $\partial p_u^{\text{priv}} / \partial p_{uk} = 0$, we have

$$p_u = \left(R_u Q - b_u Q - b_i Q - \mu Q - \left(|R_u|^{-1/2} \sum_{j\in R(u)} y_j\right) Q^T Q - \frac{1}{N}\mathbf{b}\right) \times \left[Q^T Q + \left(\lambda n_u + \frac{1}{2}\Delta\right) I\right]^{-1}. \tag{18}$$

Similarly, given a fixed P, when $\forall 1 \le i \le m$, we can solve Q as follows:

$$q_i = \left(R_i P - b_u P - b_i P - \mu P - \left(|R_i|^{-1/2} \sum_{j\in R(i)} y_j\right) P^T P - \frac{1}{N}\mathbf{b}\right) \times \left[P^T P + \left(\lambda n_i + \frac{1}{2}\Delta\right) I\right]^{-1}, \tag{19}$$

where $n_i = |R_i|$, $R_i = \{r_{uv} \in R \mid v = i\}$.

Theorem 5. *Given the differential privacy parameter ε and the parameter for computing the slack term C, if $\|p_u\|^2$, $\|q_i\|^2$, and the loss functions of ALS are convex and differentiable, Algorithm 2 provides ε-differential privacy.*

Proof. Our Algorithm 2 satisfies the application condition of Algorithm 2 in [20], which was proven to provide ε-differential privacy; thus our Algorithm 2 also provides ε-differential privacy.

Another privacy-preserving ALS algorithm of SVD++ is the ALS output perturbation method, which is shown in Algorithm 3.

In the objective function of ALS (i.e., formula (11)), each user vector p_u and item vector q_i can be obtained by solving the following risk minimization problem:

$$\begin{aligned} p_u(R,Q) &= \arg\min_{p_u} J_Q(p_u,R), \\ q_i(R,P) &= \arg\min_{q_i} J_P(q_i,R). \end{aligned} \tag{20}$$

The main idea of Algorithm 3 is that it guarantees DP by adding a random noise vector b to the output of $p_u(R,Q)$ and $q_i(R,P)$. □

Regarding Algorithm 3, a few explanatory points should be stated as follows:

(1) $p_{\max}$ and $q_{\max}$ are the upper bounds on $\|p_u\|^2$ and $\|q_i\|^2$, respectively; $\Delta r = r_{\max} - r_{\min}$. Because $p_u(R,Q)$ and $q_i(R,P)$ are the L_2-sensitivity values, their global sensitivities can be obtained as $\mathrm{GS}p_u = 2q_{\max}\Delta r/n_u\lambda$ and $\mathrm{GS}q_i = 2p_{\max}\Delta r/n_i\lambda$.

(2) According to the Laplace mechanism, for a fixed matrix Q, a random noise vector b with the pdf $f(b) \propto \exp(-\varepsilon\|b\|/2k \cdot n_u\lambda/2q_{\max}\Delta r)$ is generated. For a fixed matrix P, a random noise vector b with the pdf $f(b) \propto \exp(-\varepsilon||b||/2k \cdot n_i\lambda_1/2p_{\max}\Delta r)$ is generated.

(3) For the ALS objective function of SVD++ (formula (11)), we have Corollary 6 and Theorem 7 as follows.

Corollary 6. *Let r_{ui} refer to the rating of user u for item i. The predictive rating in SVD++ is $\tilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2}\sum_{j\in R(u)} y_j)$. $N(\cdot) = \|p_u\|^2$ is differentiable and 1-strongly convex and the loss function $\ell = (r_{ui} - \tilde{r}_{ui})^2$ is convex and differentiable with $|\ell'(\cdot)| \le 1$. Then, the L_2-sensitivity of $J_Q(p_u,R)$ is at most $2q_{\max}\Delta r/n_u\lambda$.*

Proof. Let there be two rating matrices that differ in the value of the last entry:

$$R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{pmatrix}, \quad R' = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r'_{nm} \end{pmatrix}. \tag{21}$$

Moreover, let

$$\begin{aligned} G(p_u) &= J_Q(p_u,R), \\ g(p_u) &= J_Q(p_u,R') - J_Q(p_u,R), \\ p_{u1} &= \arg\min_{p_u} J_Q(p_u,R), \\ p_{u2} &= \arg\min_{p_u} J_Q(p_u,R'), \\ g(p_u) &= \left(r'_{ui} - \tilde{r}_{ui}\right)^2 - \left(r_{ui} - \tilde{r}_{ui}\right)^2. \end{aligned} \tag{22}$$

Second, due to the convexity of ℓ and the 1-strongly convexity of $N(\cdot) = \|p_u\|^2$, $G(p_u) = J_Q(p_u,R)$ is $n_u\lambda$-strongly convex.

In addition, due to the differentiability of $N(\cdot) = \|p_u\|^2$ and ℓ, $G(p_u)$ and $g(p_u)$ are also differentiable at all points. Then, we have

$$\begin{aligned} \nabla g(p_u) &= -2\left(r'_{ui} - \tilde{r}_{ui}\right)q_i + 2\left(r_{ui} - \tilde{r}_{ui}\right)q_i \\ &= 2q_i\left(r_{ui} - r'_{ui}\right) = 2q_i\Delta r. \end{aligned} \tag{23}$$

Then, the equation $\|\nabla g(p_u)\| = 2\Delta r\|q_i^T\| \le 2q_{\max}\Delta r$ can be obtained. Hence, the L_2-sensitivity of $J_{Q(P_u,R)}$ is less than or equal to $2q_{\max}\Delta r/n_u\lambda$. The proof now follows by an application of Lemma 1 of [20].

Similarly, the L_2-sensitivity of $q_i(R,P)$ is at most $\mathrm{GS}q_i = 2p_{\max}\Delta r/n_i\lambda$. □

Theorem 7. *Let r_{ui} refer to the rating of user u for item i. The predictive rating in SVD++ is $\tilde{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot (p_u + |R(u)|^{-1/2}\sum_{j\in R(u)} y_j)$. $N(\cdot) = \|p_u\|^2$ and $N(\cdot) = \|q_i\|^2$ are differentiable and 1-strongly convex and the loss function $\ell = (r_{ui} - \tilde{r}_{ui})^2$ is convex and differentiable with $|\ell'(\cdot)| \le 1$. Then, Algorithm 3 provides ε-differential privacy.*

Proof. The proof of Theorem 7 follows from Corollary 6 and [20].

(1) According to the proof of Corollary 6, if the conditions on $N(\cdot) = \|p_u\|^2$ and the loss function ℓ hold, the L_2-sensitivity of $J_Q(p_u,R)$ with the regularization parameter $n_u\lambda$ is at most $2q_{\max}\Delta r/n_u\lambda$.

(2) When $\|b\|$ is picked from the distribution $\nu(b) = (1/\alpha)e^{-\beta\|b\|}$, where $\beta = n_u\lambda\varepsilon/2q_{\max}\Delta r$, for a specific vector $b_0 \in \mathbb{R}^d$, the density at b_0 is proportional to $e^{-\beta\|b_0\|}$.

(3) Let $R_{n\times m}$ and $R'_{n\times m}$ be any two rating matrices that differ in the value of the last entry. Then, for any p_u, we have $g(p_u \mid R)/g(p_u \mid R') = \nu(b_1)/\nu(b_2) = e^{-(n_u\lambda\varepsilon/2q_{\max}\Delta r)(\|b_1\|-\|b_2\|)}$, where b_1 and b_2 are the corresponding noise vectors and $g(p_u \mid R)$ ($g(p_u \mid R')$, resp.) is the density of the output of Algorithm 3 at p_u when the input is R (R', resp.).

(4) If p_{u1} and p_{u2} are the respective solutions to nonprivate regularized $J_Q(\cdot)$ when the inputs are R and R', then $b_1 - b_2 = p_{u1} - p_{u2}$. From Corollary 6 and using the triangle inequality, $\|b_1\| - \|b_2\| \le \|b_1 - b_2\| \le \|p_{u1} - p_{u2}\| \le 2q_{\max}\Delta r/n_u\lambda$.

Moreover, by symmetry, the densities of the directions of b_1 and b_2 are uniform. Therefore, by construction, $\nu(b_1)/\nu(b_2) \le e^{\varepsilon}$.

(5) When fixing the latent matrix P and optimizing Q, the proof process is similar. Thus, according to the

definition of DP, Algorithm 3 provides ε-differential privacy. □

5. Experiments

5.1. Experiment Datasets. In the experiments, two datasets are used to verify that our algorithms fit not only a single kind of dataset. One dataset is a MovieLens-1M dataset from http://grouplens.org/datasets/movielens/. The other is a partial Netflix dataset (called Netflix-1M in this paper) that was captured from http://www.netflixprize.com/, which was constructed to support participants in the Netflix Prize. Some statistical properties of the selected MovieLens-1M and the Netflix-1M datasets are shown in Table 1.

5.2. Evaluation Measurement and Experimental Settings. As a frequently used methodology in machine learning and data mining, tenfold cross-validation to train and evaluate the performance of our algorithms is used. The validation datasets are divided into training and test sets with an 80/20 ratio. Then, the Root Mean Square Error (RMSE) metric is used to measure the accuracy of the predicted ratings $\tilde{r}_{ui}$. The smaller the RMSE, the more accurate the prediction is. The RMSE is computed by RMSE $= \sqrt{\sum_R (r_{ui} - \tilde{r}_{ui})^2 / |R|}$, where $|R|$ denotes the number of effective ratings; the ratings here are valid, and missing scores are not included. Considering the possible discrepancies resulting from the addition of noise, the final RMSE is averaged across multiple runs.

The selection of the parameters in each algorithm is introduced briefly.

(i) Except for Figure 4, the number of factors was set to $d = 5$.

(ii) The learning rate was set to $\gamma = 0.001$.

(iii) The regularization parameter of SVD++ was set to $\lambda = 0.125$ by cross-validation.

(iv) The number of iterations was set to $k = 20$ when the error variety is less than 0.0001.

(v) To compare with [9], the values of $p_{\max}$ and $q_{\max}$ in Algorithm 3 were set to the same values as in [9]; that is, $p_{\max} = 0.4$ and $q_{\max} = 0.5$.

(vi) The regularization parameters used to compute the user bias, item bias, and implicit feedback information were set to $\lambda_1 = 10$, $\lambda_2 = 25$ and $\lambda_3 = 10$, respectively, by referring to [1].

5.3. Experimental Results and Comparison

5.3.1. Experimental Results and Analysis. The meanings of the notation used to present the experimental results are shown in Table 2.

The work of [10] was an extension of [9], and several of the same algorithms are used in the two papers. Algorithm 4 of [9] and Algorithm 4 of [10] are the same (called differentially private SGD in the two papers), and Algorithm 5 of [9] and Algorithm 6 of [10] are the same (called differentially private ALS with output perturbation in the two papers).

Table 1: Statistical properties of the two datasets.

Property	MovieLens-1M	Netflix-1M
Users	6040	4996
Movies	3952	3999
Density	4.19%	0.19%
Average rating	3.5816	3.5956
Variance rating	1.2479	1.2208

Figure 1 shows how the results of our three algorithms compare with their baselines (without DP protection) on the two datasets.

From Figure 1, the RMSEs of the proposed algorithms did not deviate from their baselines. On the whole, the results of our algorithms for the MovieLens-1M dataset are better than for the Netflix-1M dataset, because the training samples of the Netflix-1M dataset are fewer and sparser than those of the MovieLens-1M dataset. Thus, it can be concluded that the predictive accuracy is closely related to the dataset size and scarcity, even when carrying out processing by DP. Particularly in Figure 1(b), the predictive accuracy of the ALS perturbation (Algorithms 2 and 3) becomes poor when $\varepsilon < 0.01$ and the ALS output perturbation performs worse than the other algorithms. This is mainly because it perturbs the latent factor matrices after decomposition, and the smaller the value of ε, the more noise added; as a result, the inner product of the two latent factors deviates greatly from its true value. In addition, the two ALS perturbation algorithms are better than the SGD gradient perturbation algorithm (Algorithm 1) when $\varepsilon > 0.01$, even though they were both processed by DP. Particularly, the ALS objective perturbation obtains the best predictive accuracy on the MovieLens-1M dataset, regardless of whether the privacy parameter ε is large or small; that is, the results of this approach processed by DP are the most stable. This is because the update at each iteration of SGD is significantly related to the error and each iteration of ALS is directly related to the training dataset, which means that the ALS method itself is better than SGD.

To increase the predictive accuracy, as the derivative model of SVD, SVD++ introduces implicit feedback information, such as which movies a user has evaluated in the past. Figure 2 shows the results of comparing SVD++ with SVD using three DP protection algorithms. From Figure 2, it can be seen that SVD++ provides a slightly higher advantage over SVD when using the three DP protection algorithms. Overall, the RMSE of ALS with objective perturbation is optimal, especially when $\varepsilon > 0.01$.

In addition, Figure 3 shows the results of our algorithms compared with those of the correlative algorithm of [9] on the two datasets.

In [9], Berlioz et al. also proposed SGD perturbation (called PSGD in our experiments) and ALS output perturbation (called PALS). However, they needed to do some DP preprocessing of the input matrix. In fact, preprocessing of the original input matrix, that is, adding noise to it, will affect the result of SVD++. However, our algorithms not only omit the preprocessing steps but also obtain better prediction accuracies on the two test datasets (from Figure 3). Particularly, the advantage of our ALS with objective perturbation is more obvious. Furthermore, from Figure 3, it

Table 2: The meanings of the notation used to present the experimental results.

Name	Meaning
SGDBase++	Without DP protection, no preprocessing, SGD for SVD++
ALSBase++	Without DP protection, no preprocessing, ALS for SVD++
PSGD	Algorithm 4 of [9] or Algorithm 4 of [10], with preprocessing, SGD for MF
PALS	Algorithm 5 of [9] or Algorithm 6 of [10], with preprocessing, ALS for MF
DPSS	No preprocessing, SGD gradient perturbation for SVD (refer to our Algorithm 1)
DPSAObj	No preprocessing, ALS objective perturbation for SVD (refer to our Algorithm 2)
DPSAOut	No preprocessing, ALS output perturbation for SVD (refer to our Algorithm 3)
DPSS++	Our Algorithm 1, no preprocessing, SGD gradient perturbation for SVD++
DPSAObj++	Our Algorithm 2, no preprocessing, ALS objective perturbation for SVD++
DPSAOut++	Our Algorithm 3, no preprocessing, ALS output perturbation for SVD++

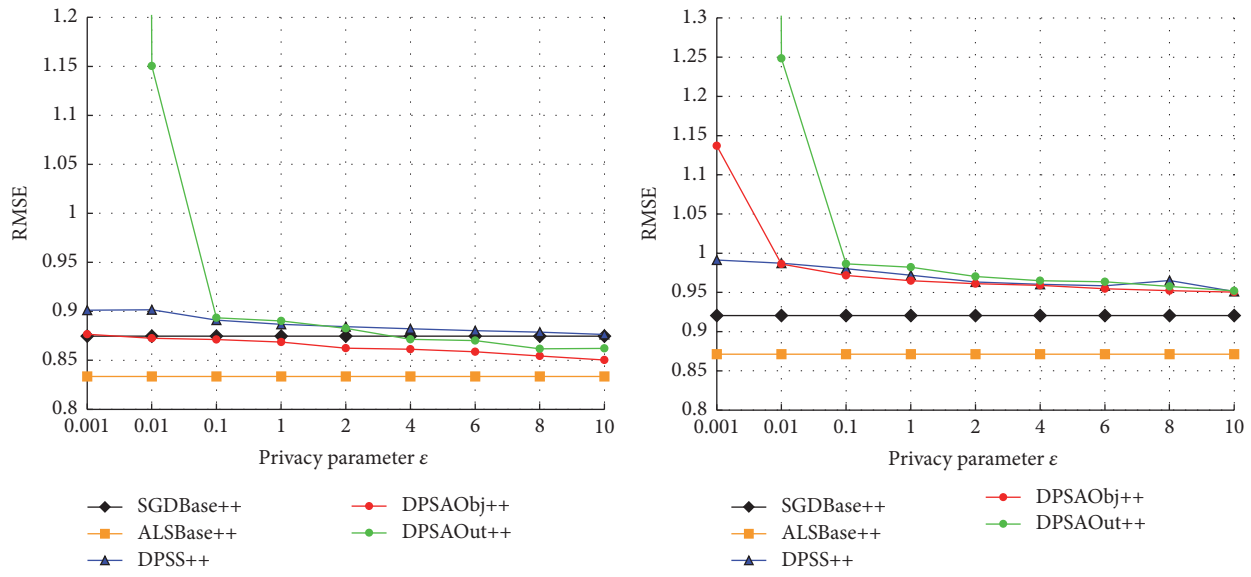


Figure 1: Comparison of the algorithm results with their respective baselines.

is worth noting that their algorithms cannot achieve better prediction accuracy when the value of ε is larger (up to 20). Moreover, the value of ε is too large and would be unreasonable according to the meaning of DP.

In addition, not only are the recommendation results of SVD++ better than those of SVD on a real dataset but also the predictive accuracy will be improved with an increase in the number of features (also called factors) in SVD and SVD++ [24]. To verify that our DP protection algorithms still have this characteristic, Figure 4 shows the relationship between the predictive accuracy and the number of factors after performing SGD gradient perturbation and ALS objective perturbation for SVD and SVD++.

In summary, the three DP algorithms that we have proposed for SVD++ can protect the privacy of the original data on the basis of ensuring the predictive accuracy. In particular, the ALS objective perturbation for the SVD++ algorithm gives a better trade-off between privacy and recommendation accuracy.

5.3.2. A Selection Scheme for DP Parameter ε. In DP applications, the strength of privacy protection depends on the parameter ε, but it is equally important to ensure the predictive accuracy when DP is applied to collaborative filtering, so a scheme for selection of DP protection parameter ε is proposed in order to balance the strength of privacy protection and the predictive accuracy. The specific steps are described as follows.

Step 1. Determine the recommended target user u.

Step 2. Compute the recommended-item set (in this paper, a movie set is used) to the user u from two aspects. Let S_1 be the recommended-item set after performing a certain DP process, and let S_2 be the recommended-item set without performing any DP process.

Step 3. Compute the intersection of the two recommended-item sets obtained in the second step, and denote it as $S = S_1 \cap S_2$.

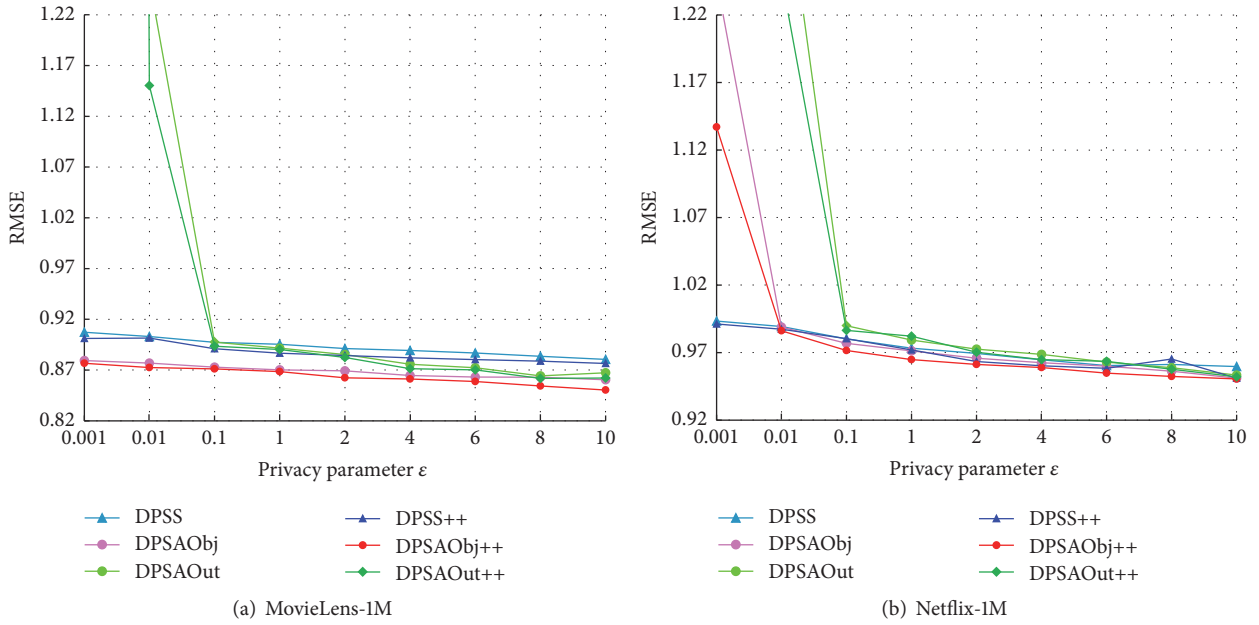


Figure 2: Comparison of SVD++ with SVD using three DP protection algorithms.

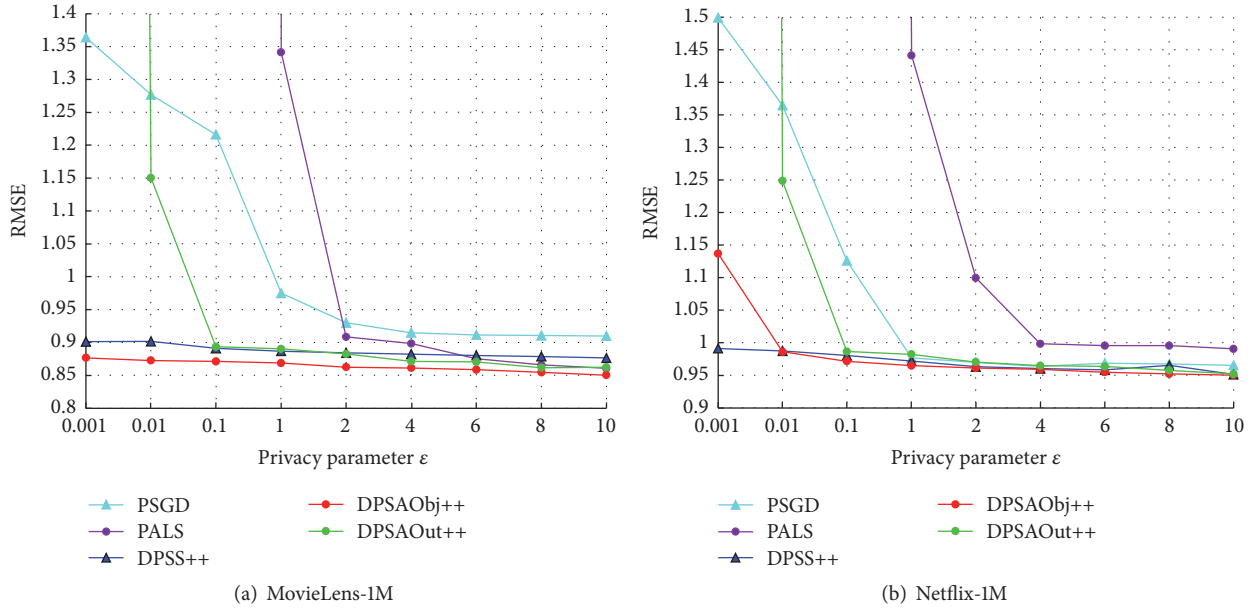


Figure 3: Comparison of our algorithms with the correlative algorithm of [9].

Step 4. If N is the total number of recommended-item sets, obtain a percentage: $P = S/N * 100\%$. The greater P is, the smaller the influence of predictive accuracy is, and the value of ε should be reasonable at this time.

This scheme can only provide a reasonable range for DP parameter ε. Normally, if this percentage is less than 20%, the recommended results are considered to be seriously affected, even though the privacy protection is very strong. On the other hand, if this percentage is more than 80%, the power of privacy protection is thought to be too weak, even though the recommendation results are better. Therefore, the value of DP parameter ε is reasonable when this percentage is between 20 and 80%. To verify this scheme, the ALS DP processes of SVD, SVD++, and the correlation algorithm of [9] (PALS) are compared, and Figure 5 shows the impact of DP parameter ε on the MovieLens-1M dataset. Each parameter in this experiment is still set in accordance with the description given in Section 5.2. In addition, the number of recommended-movie sets is set to 30 and the recommended user is selected randomly. At the same time, the result is the average value of ten runs because of the randomness of Laplace noise.

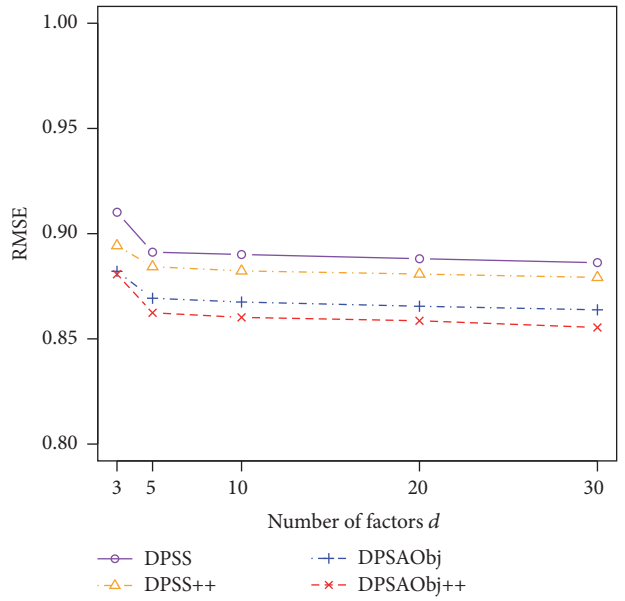


Figure 4: The relationship between the accuracy of recommendation and the number of factors.

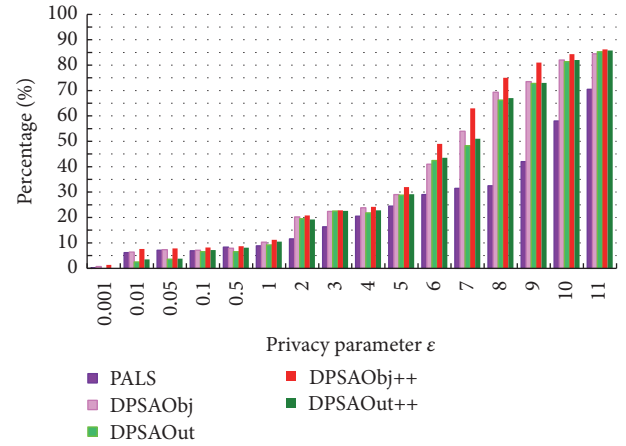


Figure 5: Comparison of the impacts of privacy parameter ε on the recommendation results.

From Figure 5, it can be concluded that the impacts of the privacy parameter ε on the recommendation results of the three new algorithms (especially Algorithm 2) are smaller than those for Algorithm 2 from [9] and SVD, which carries out the same process using DP. For our two algorithms, the coincidence degree of the recommended-movie set is found to be between 20% and 80% when the value of the privacy parameter ε is between 2 and 11. In other words, the values of ε in this percentage range can balance the privacy strength and predictive accuracy better.

6. Discussion

Currently, the services provided by the Web are richer and more colourful. While data providers can obtain convenient personalized services and Web businesses can thus obtain more profits, which is a win-win situation. However, the leakage of personal privacy information has become a very worrying problem for many users. A variety of Internet records on users, film scores, the purchase of goods, and other information provide attackers with a certain background knowledge and personal privacy information can be derived indirectly. Therefore, in order to protect the private information of the original data on the basis of ensuring the predictive accuracy, we proposed three new methods that apply differential privacy to SVD++ through gradient perturbation, objective-function perturbation, and output perturbation. Rigorous mathematical proofs are given to ensure that all three methods maintain the differential privacy. According to experimental verification and comparison with DP privacy-preserving based on SVD and [15] on two real datasets, our new algorithms for SVD++ give better experimental results, especially the approach of ALS objective perturbation for SVD++ (Algorithm 2), which obtained better results in terms of balancing privacy and prediction. A scheme for the selection of DP parameters is finally proposed, and it can obtain a reasonable range for the DP parameter, balancing privacy, and recommendation accuracy.

Recommender systems and the field of data mining require healthy development and are inseparable from the protection of privacy in in-depth research. In the future, a more in-depth study of the following aspects can be expected.

(i) Relative parameter tuning for SVD++: typically, SVD++ parameters, such as the number of factors, the regularization parameter, and the learning rate, are tuned to increase prediction accuracy, while preventing overfitting and ensuring convergence.

(ii) More effective selection of DP parameter ε: in this paper, only the selection interval of ε is provided, but it is hard to determine the optimal ε. After all, the Laplace noise itself is random.

(iii) Comparison of other collaborative filtering or recommender algorithms: in this paper, the new approach is the application of DP to the optimal algorithms of SVD++. To extend the application of DP, other collaborative filtering or recommender algorithms could be studied and compared with one another in terms of their recommender effects.

(iv) Multiple evaluation measurements might be used to verify the new algorithms.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is sponsored in part by the Natural Science Foundation of Guangdong Province (nos. 2014A030313662 and 2016A030310018) and College Students' Science and Technology Innovation Fund of Guangdong Province (no. G2016Z08).

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, Springer, Berlin, Germany, 2010.

[2] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[3] B. Mehta, T. Hofmann, and W. Nejdi, "Robust collaborative filtering," in *Proceedings of the 1st ACM Conference on Recommender Systems (RecSys '07)*, pp. 49–56, Minneapolis, Minn, USA, October 2007.

[4] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 426–434, Las Vegas, Nev, USA, August 2008.

[5] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP '06)*, pp. 1–12, Venice, Italy, July 2006.

[6] K. Su, L. L. Ma, B. Xiao, and H. Q. Zhang, "Web service QoS prediction by neighbor information combined non-negative matrix factorization," *Journal of Intelligent and Fuzzy Systems*, vol. 30, no. 6, pp. 3593–3604, 2016.

[7] Q. Liu, Q. Wu, Y. Zhang, and X. Wang, "Recommendation-based third-party tracking monitor to balance privacy with personalization," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS '14)*, pp. 1472–1474, Scottsdale, Ariz, USA, November 2014.

[8] P. Dandekar, N. Fawaz, and S. Ioannidis, "Privacy auctions for recommender systems," *ACM Transactions on Economics and Computation*, vol. 2, no. 3, pp. 1–22, 2014.

[9] A. Berlioz, A. Friedman, M. A. Kaafar, R. Boreli, and S. Berkovsky, "Applying differential privacy to matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*, pp. 107–114, Vienna, Austria, September 2016.

[10] A. Friedman, S. Berkovsky, and M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, pp. 425–458, 2016.

[11] J. Canny, "Collaborative filtering with privacy," in *Proceedings of the IEEE Symposium on Security and Privacy (S and P '02)*, pp. 45–57, Berkeley, Calif, USA, May 2002.

[12] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 627–635, Paris, France, July 2009.

[13] T. Q. Zhu, G. Li, Y. L. Ren, W. L. Zhou, and P. Xiong, "Differential privacy for neighborhood-based collaborative filtering," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '13)*, pp. 752–759, ACM, Ontario, Canada, August 2013.

[14] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI '15)*, pp. 1763–1770, Buenos Aires, Argentina, July 2015.

[15] Z. Liu, Y.-X. Wang, and A. J. Smola, "Fast differentially private matrix factorization," in *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*, pp. 171–178, Vienna, Austria, September 2015.

[16] X. Zhu and Y. Sun, "Differential privacy for collaborative filtering recommender algorithm," in *Proceedings of the 2nd ACM International Workshop on Security and Privacy Analytics (IWSPA '16)*, pp. 9–16, New Orleans, La, USA, March 2016.

[17] S. Yan, S. Pan, W. Zhu, and K. Chen, "DynaEgo: privacy-preserving collaborative filtering recommender system based on social-aware differential privacy," in *Information and Communications Security*, vol. 9977 of *Lecture Notes in Computer Science*, pp. 347–357, Springer International, Cham, Switzerland, 2016.

[18] O. Javidbakht and P. Venkitasubramaniam, "Differential privacy in networked data collection," in *Proceedings of the Annual Conference on Information Science and Systems (CISS '16)*, pp. 117–122, Princeton, NJ, USA, March 2016.

[19] R. Balu and T. Furon, "Differentially private matrix factorization using sketching techniques," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '16)*, pp. 57–62, ACM, Vigo, Spain, June 2016.

[20] K. Chaudhuri, C. Monteleoni, and A. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, pp. 1069–1109, 2011.

[21] C. Dwork, F. F. McShery, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Conference on Theory of Cryptography (TCC '06)*, pp. 265–284, New York, NY, USA, March 2006.

[22] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the 39th Annual ACM Symposium on Theroy of Computing (STOC '07)*, pp. 75–84, San Diego, Calif, USA, June 2007.

[23] M. Hardt and K. Talwar, "On the geometry of differential privacy," in *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC '10)*, pp. 705–714, Cambridge, Mass, USA, June 2010.

[24] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 447–456, Paris, France, June 2009.

Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8698230, 9 pages
https://doi.org/10.1155/2017/8698230

Research Article

Efficient Data Transmission Based on a Scalar Chaotic Drive-Response System

Ang Li and Cong Wang

School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Cong Wang; wangc@bupt.edu.cn

Received 14 November 2016; Accepted 28 December 2016; Published 31 January 2017

Academic Editor: Liu Yuhong

Based on a scalar chaotic drive-response system, an efficient big data transmission scheme has been presented in this paper. In our method, the sender can modulate a great quantity of messages in the drive system using Walsh function, and the receiver can recover the original data using our proposed efficient reconstruction algorithm. To explore the feasibility and effectiveness, a series of simulations are performed and the results show that our proposed scheme outperforms some traditional approaches. This scheme has some potential applications in chaotic laser communication.

1. Introduction

Big data brings people much convenience as well as many problems. In the area of big data, data is the carrier of information, and the exchange of information cannot be separated from the transmission of data. Therefore, the problem of big data secure transmission becomes very serious and cannot be avoided [1, 2]. In recent years, chaotic secure communication has been one of the research focuses in the field of communication [3, 4]. Because of the remarkable contribution of Pecora and Carroll who addressed the synchronization of chaotic systems using a drive-response conception [5], the research on chaotic secure communication based on chaotic synchronization attracted wide attention and gradually infiltrated to many other subjects [6–11]. In fact, the dynamic behavior of chaotic system has some properties, such as initial sensitivity and unpredictability. These excellent properties have led to some applications of chaotic synchronization, such as chaos masking [12–14], chaos shift keying [15, 16], and chaotic modulation [17–20]. In recent years, a large number of improved chaotic communication models have emerged, such as the combination of chaos communication and multiplexing technology [21, 22], wireless chaotic communication [23], ultrawideband chaotic communication [24], chaotic laser communication [25, 26], and chaotic communication scheme based on wave recorder and time delay [27]. Recently, one significant topic of chaotic communication mainly focuses on the time series analysis [28–30]. But how many messages can be transmitted by one scalar chaotic signal? In our previous work [31], we have already achieved multiple information transmission only using one scalar chaotic time series; however, in that scheme, the original data is modulated into the system parameters directly which limits the maximum quantity of transmitted information data.

The contribution of this paper lies in the following aspects. First, a novel multiple time-delay chaotic communication scheme for big data transmission is designed based on Walsh function by which a huge amount of information can be modulated into a chaotic system. Specifically, the sender integrates multiple original information into single information by using Walsh function and then modulates such integrated information into the parameters of the drive system. Next, we design an adaptive parameter estimation scheme to recover the integrated information. That is to say, the receiver can use the inverse mapping of Walsh function to recover the original information. At last we investigate the maximum amount of information carried by a scalar chaotic drive-response system. Based on Shannon's channel capacity theorem, because of the channel bandwidth and noise, there

exists a boundary of the maximum information in a real communication channel [32, 33]. To explore the boundary of maximum transmittable information, we perform extensive simulations and find that our scheme is much more effective than the traditional technologies.

The remainder of this paper is structured in the following manner. We introduce the mathematical proof of the chaotic synchronization and the parameter adaptive estimation criterion in Section 2. Section 3 describes the design of chaotic communication scheme based on Walsh function and demonstrates the information recovery algorithm. In Section 4, the experimental results are showed to find out the maximum number of information carried by our scheme. Section 5 analyzes the application of our scheme. Finally, we draw our conclusions in Section 6.

Some symbols are used in this paper which are presented in Notations.

2. The Adaptive Synchronization Scheme

In this paper, we study the efficient data transmission using a scalar chaotic signal. For this purpose, we design a system model to carry as much information as possible. Based on the Mackey-Glass system [34], we consider a scalar time-delay chaotic system as follows:

$$\dot{x}(t) = -\alpha x(t) + \frac{\beta x(t-\tau)}{1 + x^{\gamma}(t-\tau)} + \sum_{i=1}^{m} a_i x(t-\tau_i), \quad (1)$$

where $x(t)$ denotes the state variable of the system, α, β, and γ are constants, and τ, τ_i are the time delays. $a_1, a_2, \ldots, a_m$ are system parameters which represent the original messages in this paper. Therefore, the bigger m is, the more information the system can carry. In this model, we can adjust the amount of information carried by the system by changing the time delays τ, τ_i.

Based on the system in (1), a communication scheme is proposed. As the information is modulated in the system parameters, we make use of the parameter estimation method to get the recovered information. Based on synchronization principle, we design the following response system and the adaptive criterion:

$$\begin{aligned} \dot{y}(t) &= -\alpha y(t) + \frac{\beta x(t-\tau)}{1 + x^{\gamma}(t-\tau)} + \sum_{i=1}^{m} \widehat{a}_i x(t-\tau_i) \\ &\quad + u(t), \\ u(t) &= -(\eta + \alpha) e(t), \\ \dot{\widehat{a}}_i &= -e(t) x(t-\tau_i), \end{aligned} \quad (2)$$

where $\widehat{a}$ is the estimated parameter, $u(t)$ is the controller, and η is a positive constant. $e(t)$ denotes the error term, which can be defined as $e(t) = y(t) - x(t)$. According to the drive system and the response system, the error system can be written as

$$\dot{e}(t) = -\eta e(t) + \sum_{i=1}^{m} (\widehat{a}_i - a_i) x(t-\tau_i). \quad (3)$$

To verify that the estimated parameter $\widehat{a}_i$ converges to the original system's parameters, we present the proof as follows.

The Lyapunov function $V(t)$ is constructed as

$$V(t) = \frac{1}{2} e^2(t) + \frac{1}{2} \sum_{i=1}^{m} (\widehat{a}_i - a_i)^2. \quad (4)$$

The time derivative of $V(t)$ along the trajectories of (4) is described as follows:

$$\begin{aligned} \dot{V}(t) &= e(t)\dot{e}(t) + \sum_{i=1}^{m} (\widehat{a}_i - a_i)(\dot{\widehat{a}}_i - \dot{a}_i) \\ &= -\eta e^2(t) + e(t) \sum_{i=1}^{m} (\widehat{a}_i - a_i) x(t-\tau_i) \\ &\quad + \sum_{i=1}^{m} (\widehat{a}_i - a_i) \dot{\widehat{a}}_i \\ &= -\eta e^2(t) + e(t) \sum_{i=1}^{m} (\widehat{a}_i - a_i) x(t-\tau_i) \\ &\quad + \sum_{i=1}^{m} (\widehat{a}_i - a_i)(-e x(t-\tau_i)) = -\eta e^2(t) \le 0. \end{aligned} \quad (5)$$

Obviously, $\dot{V} = 0$ if and only if $e = 0$. From Barbalat's lemma, we can easily get $e \to 0$ and $(\dot{\widehat{a}}_i - \dot{a}_i) \to 0$ as $t \to \infty$. Thus, we can acquire the largest invariant set M which is defined as $M = \{e \in R^n,\ (\widehat{a}_i - a) \in R^m | e = 0,\ -e + \sum_{i=1}^{m} (\widehat{a}_i - a_i) x(t-\tau_i) = 0\}$. In this case, the following equation can be satisfied:

$$\sum_{i=1}^{m} (\widehat{a}_i - a_i) x(t-\tau_i) = 0. \quad (6)$$

Let $D(x) = \{x(t-\tau_1), x(t-\tau_2), \ldots, x(t-\tau_m)\}$, $\widehat{A} = (\widehat{a}_1, \widehat{a}_2, \ldots, \widehat{a}_m)^T$, and $A = (a_1, a_2, \ldots, a_m)$. Then, (6) can be written as follows:

$$D(x)(\widehat{A} - A) = 0. \quad (7)$$

Then both sides of (7) are multiplied by $D(x)^T$ and integrated for any period of time σ, and we get the following:

$$\int_{s}^{s+\sigma} D(x)^T D(x)(\widehat{A} - A)\, dt = 0. \quad (8)$$

Let $G = \int_{s}^{s+\sigma} D^T(x(t)) D(x(t)) dt$. G is called the Gram matrix of $D(x)$. Then we get $G(\widehat{A} - A) = 0$. If G has full rank, (8) has a unique zero solution [35, 36]. That is to say; $\widehat{A} - A = 0$, that is, $\widehat{a}_i = a_i$. The proof of the synchronization and estimation criterion for the chaotic system is completed.

3. The Walsh-Based Transmission Scheme

In this section, we design a transmission scheme based on Walsh function which can further increase the maximum

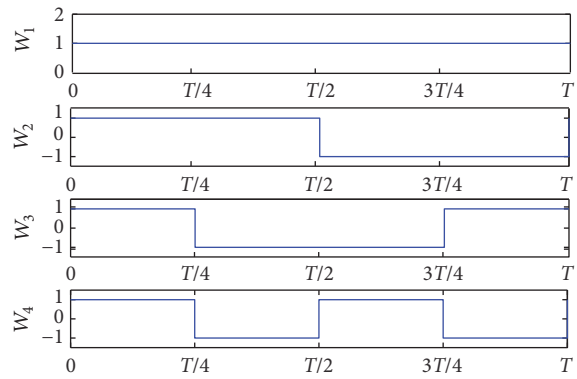


Figure 1: The 4-order Walsh Function.

quantity of transmitted information. The Walsh function is a kind of nonsinusoidal orthogonal complete function set [37]. A 4-order Walsh function is depicted in Figure 1.

It is easy to find that the elements of Walsh function set fully satisfy the orthogonality with each other. Note that as the number of available sequences is very large, it satisfies the demand of multiple information transmission.

Based on the properties of Walsh function, we consider a system based on the Mackey-Glass system; the drive system (1) can be redesigned as follows:

$$\dot{x}(t) = f(x(t)) + \sum_{i=1}^{m}\sum_{j=1}^{k}\left(a_{ij}W_j(t)\right)x(t-\tau_i), \tag{9}$$

where $f(x(t)) = -\alpha x(t) + \beta x(t-\tau)/(1+x^{\gamma}(t-\tau))$, a_{ij} is the transmitted original message, and $W_j(t)$ is the jth Walsh function among k-orders Walsh function. In this way, there are k original messages in each system parameter. Therefore, the number of message increases from m to $m \times k$.

We introduce the following formula to measure the total number of messages carried by this scheme:

$$Q = \frac{H}{l_b}(m \times k), \tag{10}$$

where Q denotes the quantity of total information (bits) carried by the system, H is the effective length of the carrier, l_b represents the length of one bit of information, and m and k are the number of the system parameters and the orders of Walsh function, respectively.

The corresponding response system and the adaptive criterion can be designed as follows:

$$\begin{aligned}
\dot{y}(t) &= f(y(t)) + \sum_{i=1}^{m}\widehat{b}_i x(t-\tau_i) + u(t),\\
u(t) &= -(\eta+\alpha)e(t),\\
\dot{\widehat{b}}_i &= -e(t)\,x(t-\tau_i),
\end{aligned} \tag{11}$$

where $f(y(t)) = -\alpha y(t) + \beta x(t-\tau)/(1+x^{\gamma}(t-\tau))$ and $\widehat{b}_i$ is the estimated information of the system parameters. As we already proved the synchronization of the system, similarly, the system presented in (9) can also be synchronized by following the same procedure.

Theoretically, the estimated parameters converge to the true value when $t \to \infty$. However, in practical scenarios, it requires a very short time. More precisely, the estimated parameters take a transient time to approach the true values and after that they remain unchanged. Thus, if we set up a sampling point at each unchanged period and then design a threshold mechanism to distinguish the estimated parameters, we get the estimated system parameters precisely. Based on (11), as a_{ij} is binary, thus $a_{ij}W_j(t)$ must be integral; the threshold mechanism can be designed as follows:

$$\begin{aligned}
&\text{output} = n, \quad \text{if } \frac{2n-1}{2} \le \widehat{b}_i\Big|_{t_j} < \frac{2n+1}{2};\\
&\text{output} = 0, \quad \text{if } -0.5 \le \widehat{b}_i\Big|_{t_j} < 0.5;\\
&\text{output} = -n, \quad \text{if } \frac{-2n-1}{2} \le \widehat{b}_i\Big|_{t_j} < \frac{-2n+1}{2},
\end{aligned} \tag{12}$$

where $n = 1, 2, 3, \ldots, k$, $t_j = jl_w$ is the sample time and $l_w = l_b/k$ denotes the length of k-orders Walsh function's symbol. Until the convergent time remains short enough for the threshold mechanism, we get $\widehat{b}_i = \sum_{j=1}^{k} a_{ij}W_j(t)$.

Next, we present the recovering algorithm of the Walsh function to recover the original information. We multiply $\widehat{b}_i$ by the corresponding Walsh function then integrate them for each period T and thereby the original message is recovered. For example, if the information to be recovered is a_{pq} ($1 \le p \le m$, $1 \le q \le k$), then the estimated information is $\widehat{b}_p$. As we proved before, $\widehat{b}_p = \sum_{j=1}^{k} a_{pq}W_j(t)$. The process of calculation is presented as follows:

$$\begin{aligned}
\int_{(\theta-1)T}^{\theta T}\widehat{b}_p W_q(t)\,dt &= \int_{(\theta-1)T}^{\theta T}\left[\sum_{j=1}^{k}a_{pq}W_j(t)\right]W_q(t)\,dt\\
&= a_{pq}\int_{(\theta-1)T}^{\theta T}W_q(t)^2\,dt\\
&\quad + a_{pq}\sum_{j=1,\,j\neq q}^{k}\int_{(\theta-1)T}^{\theta T}W_j(t)\,W_q(t)\,dt = a_{pq},\\
&\qquad (\theta = 0, 1, 2, 3, \ldots).
\end{aligned} \tag{13}$$

Remark 1. Step 2 and step 3 of (13) are using the property of Walsh function that

$$\int_0^T W_i(t)\,W_j(t)\,dt = \begin{cases} 0, & i \neq j,\\ 1, & i = j, \end{cases} \quad (i, j \in k). \tag{14}$$

As a result $\int_{(\theta-1)T}^{\theta T}W_q(t)^2 = 1$ and $\int_{(\theta-1)T}^{\theta T}W_j(t)W_q(t) = 0$.

Thus, a chaotic communication model that combines the Walsh function and the adaptive parameter identification technique is finally obtained. Thus far, the Walsh-based transmission scheme has been established. The main process is presented in Figure 2.

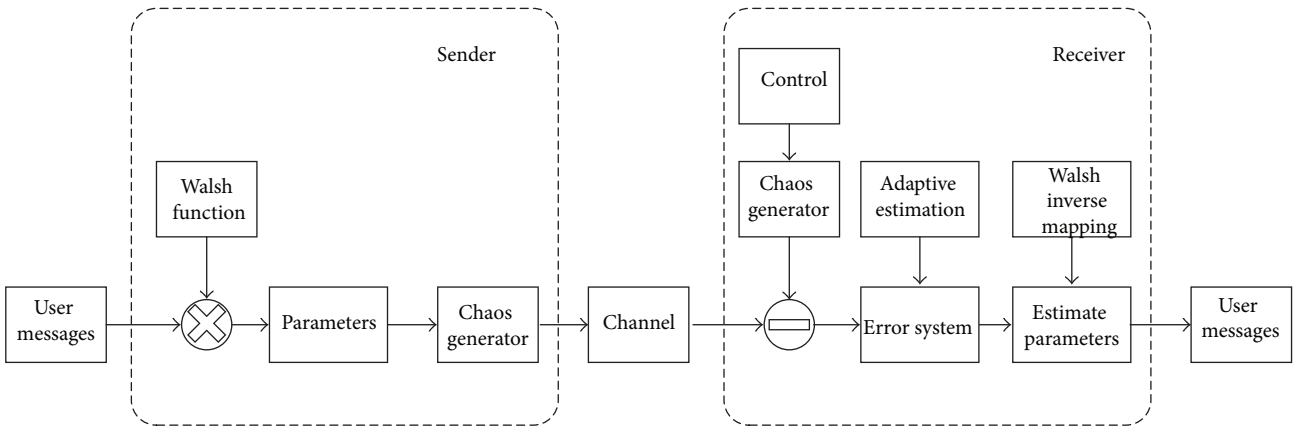


Figure 2: The flowchart of a procedure of communication.

Remark 2. We present some comparisons between different communication schemes on the total amount of messages. First, in our scheme, plenty of messages can be made into one mixed message, furthermore, many such mixed messages be can modulated into a multiple time-delay system; thus in our scheme the quantity of messages carried by the system is very huge ($m \times k = 960$). In the chaos masking scheme, only one carrier of message is carried by the chaotic system, that is, $m = 1$. In the chaotic modulation scheme, the value of m depends on the system's dimensions as the messages are modulated into the system; thus the chaotic system will be very complex. In the chaotic shift keying scheme, m equals the number of the system parameters which steal less than ours. Compared with these communication schemes, our scheme strongly increases the total amount of messages carried by the chaotic system. In addition, our scheme uses a scalar chaotic signal which makes it easier to produce and transmit.

4. Experiment and Simulation

In this section, we will explore the maximum quantity of transmitted information by our scheme. At first, we consider a system based on the Mackey-Glass model as presented below:

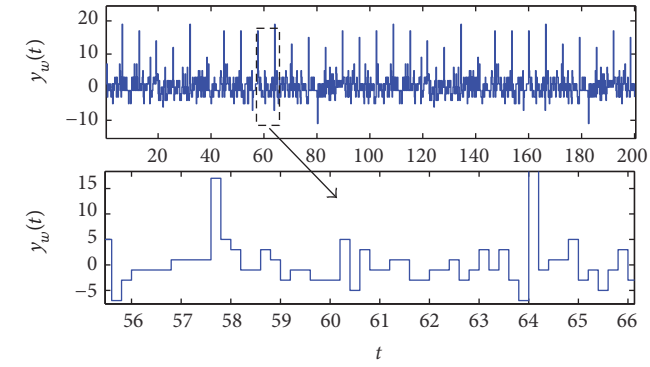
$$\begin{aligned} \dot{x} &= f(x(t)) + \sum_{i=1}^{m}\sum_{j=1}^{k}\left(a_{ij}W_j(t)\right)x(t-\tau_i), \\ \dot{y} &= f(y(t)) + \sum_{i=1}^{m}\widehat{b}_i x(t-\tau_i), \end{aligned} \tag{15}$$

where $f(x(t)) = -1100x(t) + 50000x(t-\tau)/(1+x^{20}(t-\tau))$, $f(y(t)) = -1100y(t) + 50000x(t-\tau)/(1+x^{20}(t-\tau))$, and $\alpha = -1100$, $\beta = 50000$, $\gamma = 20$, $\tau = 0.5$, $\tau_i = 1+0.1i$. a_{ij} is the original information represented as random binary sequence with arbitrary length. In the simulation, we set up the relative tolerance to $1 \times e^{-4}$.

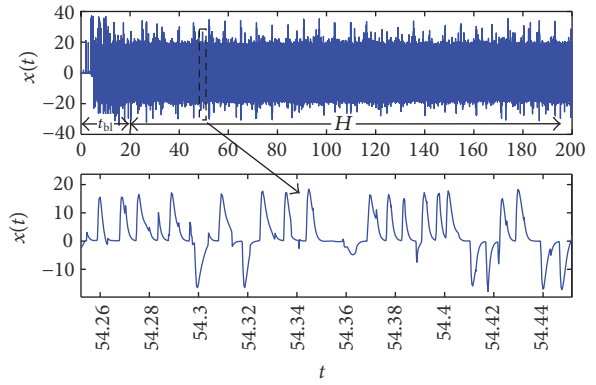
Remark 3. To ensure the chaotic property of the system, we attempt to adjust the values of α, β, and γ appropriately. We have set different values of α, β, and γ to start simulation, and at last we find the system has an excellent chaotic property when $\alpha = -1100$, $\beta = 50000$, $\gamma = 20$, and $\tau = 0.5$.

4.1. Simulation with Different m. As the quantity of the transmitted information is determined by $m \times k$, we first choose $k = 32$; that is, we use the 32-order Walsh function. Subsequently, we increase m as required. For $m = 20$, the corresponding results are shown in Figures 3(a)–3(f). Figure 3(a) displays the information combined by Walsh function. It forms an integral wave. The length of each bit is set to 0.2; that is, $l_w = 0.2$. For making the original binary information to satisfy the orthogonal relation, the bit width of the original information is set to 6.4; that is, $l_b = 6.4$. Since there is a block time $t_{\text{bl}} = 20$ for the running system from the initial state to the stable state, we cannot recover the information until $t \geq 20$; the effective length of the scalar series is taken as $H = 180$. As mentioned before, the number of the system parameters is selected as $m = 20$ and the order of Walsh function as $k = 32$; thus, based on (10), the quantity of information loaded in the system is $Q = 18000$.

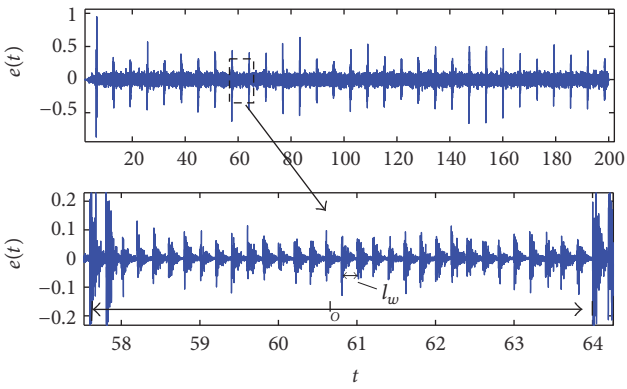
Figure 3(b) shows that a scalar chaotic signal $x(t)$ is sent by the sender. Based on chaotic synchronization, we get the error signal as depicted in Figure 3(c). We observe that the synchronization error will converge to 0 for each sampling time ($t_j = 0.2$) from the details of $e(t)$. Hence, the estimated values converge to the value integrated by Walsh function in each sampling time as presented in Figure 3(d). We set up a sampling point at $t = 0.2j$. In this way, we can accurately estimate the accurate Walsh integrated information. After that, based on (13), we let the estimated value $\widehat{b}_i$ be multiplied with the corresponding Walsh function and then integrate them in one period of Walsh function. If the obtained original binary information is 1, the result of the integral will be positive; otherwise, the result of integral will remain unchanged. Thus, we get a ladder-like waveform as presented in Figure 3(e). From that ladder-like waveform, we can recover the original binary information by using the



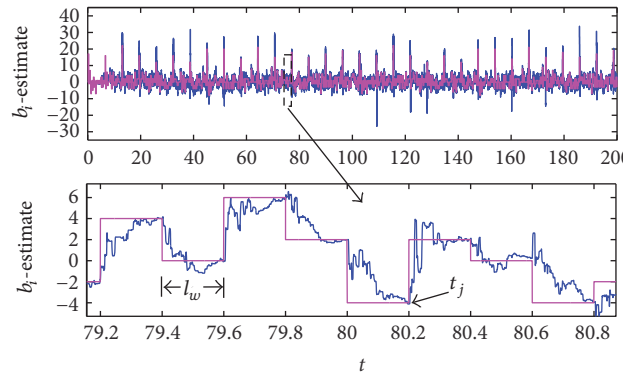
(a) The Walsh integrated message. $y_w(t) = \sum_{j=1}^{k} a_{ij} W_j(t)$



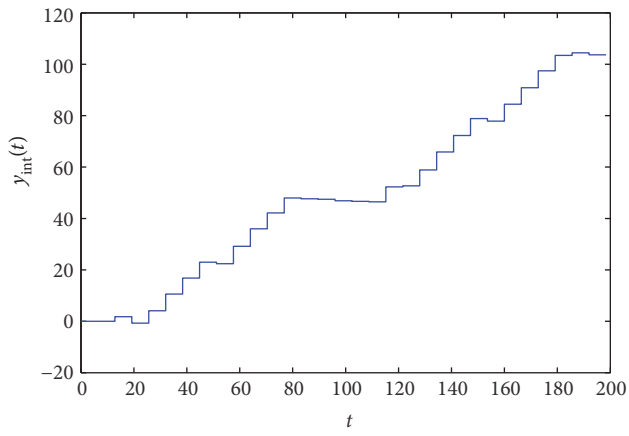
(b) The transmitted signal $x(t)$



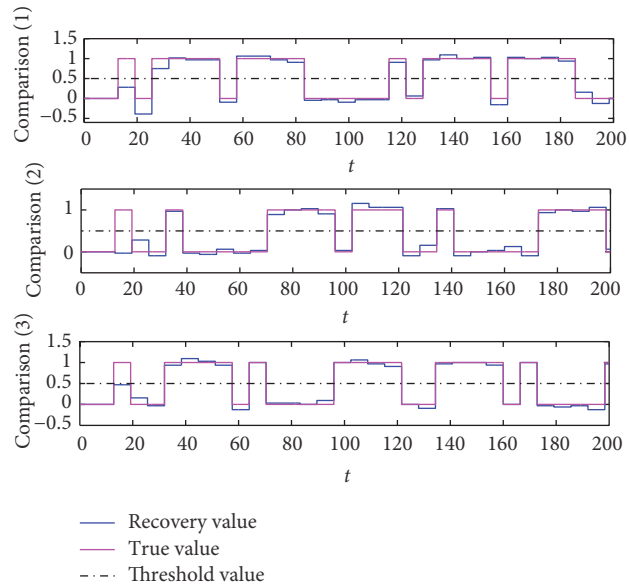
(c) The system error $e(t)$ when $m = 20$ *and* $k = 32$



(d) The stochastic comparison of estimated value and the accurate value of the 20 Walsh integrated messages



(e) The ladder-like trance of integral. $y_{\text{int}}(t) = \int_{(\theta-1)T}^{\theta T} \widehat{b}_i W_j(t) dt$. Each rising edge denotes the "1" and others denote the "0"

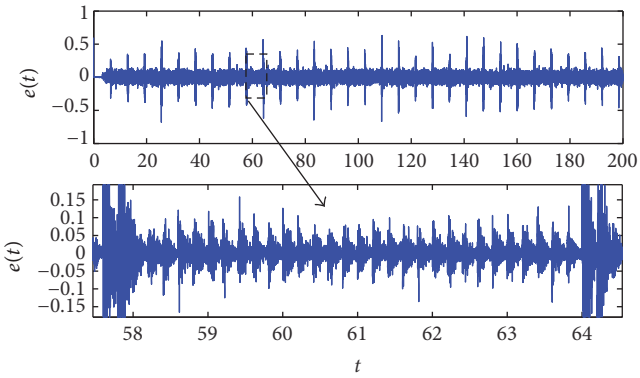


(f) The stochastic three comparisons of recovered value and original value

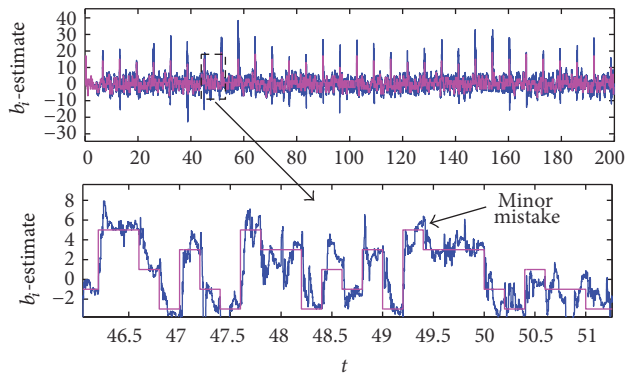
FIGURE 3: The simulation results when $k = 32$ and $m = 20$.

method that each rising edge equals "1" and others equal "0." The comparisons of the recovered value and the original value are shown in Figure 3(f). We set a threshold a_{th} ($a_{\text{th}} = 0.5$); we can easily distinguish 0 and 1. Thus, the transmitted information is precisely restored.

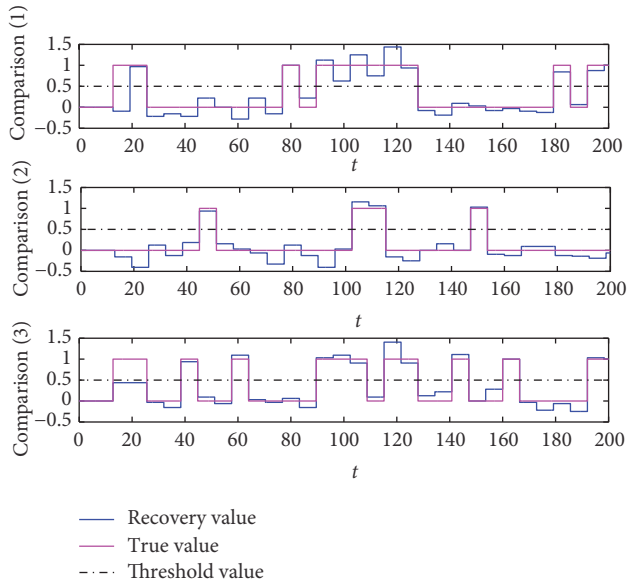
In the next step, we raise the value of m to 30. The results are depicted in Figures 4(a)–4(c). The error signal in Figure 4(a) is compared with Figure 3(c). It is obvious that the rate of convergence when $m = 30$ is slower than that of when $m = 20$. Thus, it points out to some minor



(a) The system error $e(t)$ when $m = 30$ *and* $k = 32$



(b) The stochastic comparison of estimated value and the accurate value of the 30 Walsh integrated messages



(c) The stochastic three comparisons of recovered value and original value

FIGURE 4: The simulation results of $k = 32$ and $m = 30$.

mistake in Figure 4(b). This minor mistake lies within the permitted sphere of estimation when $m = 30$, so we can still recover the original information accurately. While compared with Figure 3(f), we find that the recovered information is far away from original value even almost beyond the threshold as presented in Figure 4(c). On the other hand, the recovered information lies near to the original value when $m = 20$. With the increment of m, more and more errors appear in $\widehat{b}_i$ which becomes the hurdle to recover the original information. Under the premise of the accuracy, as a result, the experimental maximum of m is 30. Thus, based on (10), the maximum quantity of information carried by the system is $Q = 27000$. This quantity of information is much larger than that of traditional chaotic communication schemes.

4.2. Simulation with Different k. Next, we change the order of the Walsh function while fixing the width of original information to $l_b = 6.4$ which is the same as $k = 32$.

TABLE 1: The total information for different k.

k	8	16	32	64
l_w	0.8	0.4	0.2	0.1
l_b	6.4	6.4	6.4	6.4
$m_{\max}$	115	58	30	14
H	180	180	180	180
Q (bit)	25875	26100	27000	25200

Under the premise that the system can accurately recover the original information, we let $k = 8, 16, 64$ and simulate the experiments for each case separately. The results are presented in Table 1.

We expect the system can carry information as much as possible, but we observe from Notations that m decreased as k increases. Thus, we cannot increase m and k at the same time. Meanwhile, the total information presents a small uptrend when $l_w \geq 0.2$ and then goes down. Thus we get the

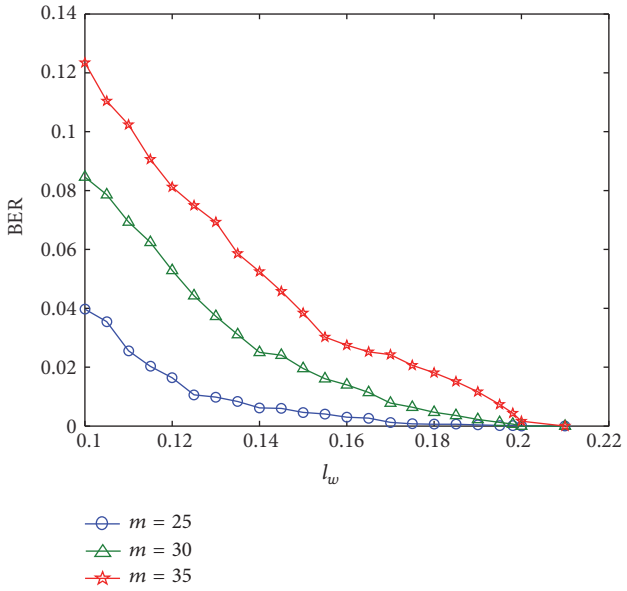


FIGURE 5: The bit error rate (BER) in different l_w and m.

maximum information when $k = 32$, and the total number of information is 27000. That is the reason to set $k = 32$ for the simulation at the beginning of this section.

Remark 4. Why choose $l_w = 0.2$? To explain this question, we perform a series of simulations with different l_w under 32-order Walsh function. The result is depicted in Figure 5. We observe from here that the BER decreases with the increment of l_w. Thus, the smaller l_w is, the more information the system could carry. We expect l_w to be as small as possible, but it should be long enough so that the estimated value can converge to the true value. Thus, under the condition of non-BER, the minimum of l_w is set to 0.2.

4.3. Simulation with Gaussian White Noise. Next, the effect of noise is under consideration in our system. We add an Gaussian white noise in the drive system which can be written as follows:

$$\dot{x}(t) = f(x(t)) + \sum_{i=1}^{m}\sum_{j=1}^{k}\left(a_{ij}W_j(t)\right)x(t-\tau_i) + G(t), \quad (16)$$

where $G(t)$ denotes Gaussian white noise with its expectation and variance set to $(0, 25)$. The result is shown in Figure 6. Despite such noise, the simulation still recover the original information. That is to say, our system has a good ability to resist system noise. As the variance of the Gaussian white noise increases, the recovery accuracy tends to decrease. In the case when the variance exceeds 29, the nonerror recovery cannot be achieved.

5. Application Analysis

In the following section, considering the Shannon-Hartley theorem [32, 33], we analyze the relationship between the signal transmission rate and the signal power in the real channel. First, we present the formula of calculating the average power of signal S as follows:

$$S = \lim_{T\to\infty}\frac{1}{T}\int_{-\infty}^{\infty} x(t)^2\,dt. \quad (17)$$

By using the formula, we can calculate the average power of $x(t)$, when $k = 32$, and m is set to $m = 1, 10, 20, 30$. Then we let $\delta = m * k$; the relation after calculations is presented as follows: $S_{\delta=32} < S_{\delta=320} < S_{\delta=640} < S_{\delta=960}$. The Shannon-Hartley theorem describes the relationship between the upper bound for the rate of transmission of information in a real channel and the channel signal-to-noise ratio and bandwidth; thus, it indicates that different bandwidths of modern wireless systems cause different maximum throughput of single carrier. The formula to characterize the theorem is presented as follows:

$$C = B\log\left(1 + \frac{S}{N_0 B}\right), \quad (18)$$

where C denotes the information transmission rate, B is the bandwidth of the channel, and N_0 is the noise power. Given B and N_0, the rate of transmission increases with the growth of the average power of signal; that is, $C_{\delta=32} < C_{\delta=320} < C_{\delta=640} < C_{\delta=960}$. In the era of big data, chaotic laser communication has great potential for mass quantity data transmission. Based on the aforementioned analysis, we conclude that if our proposed model is applied to the real chaotic laser communication, as the number of transmission information in our scheme is much larger than the traditional chaotic technology under the same setup, the efficiency of chaotic laser communication can be improved. In recent years, the long-haul and low-cost chaotic optical secure communications with 1.25 Gbits/s-message and 2.5 Gbits/s-message are experimentally realized using discrete optical components. The transmission distance reaches 143 km and 25 km [38], which is based on chaotic masking. Since the transmission rate $C_{\delta=32} < C_{\delta=960}$ under the same position, if our technology is applied in the above real system, the overall rate can be further increased to some extent; we will discuss the related issues in future research.

6. Conclusion

In summary, for the purpose of big data transmission, an efficient chaotic communication scheme based on Walsh function is designed. Experimental simulations are performed to explore the maximum value of information carried by one-dimensional scalar chaotic signal and illustrate the feasibility of this scheme. Finally, the application is discussed and will be further studied in our future works.

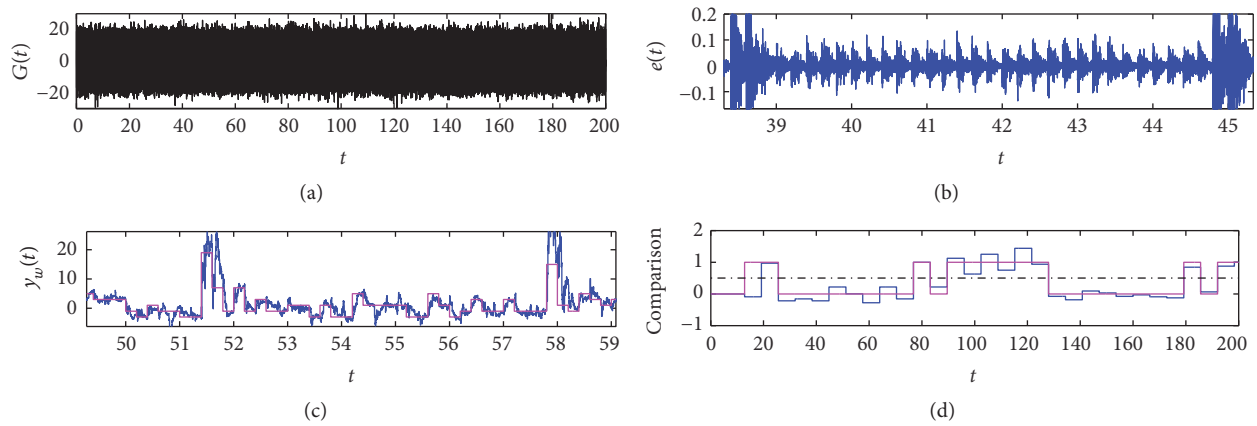


FIGURE 6: The simulation results of simulation with Gaussian white noise. (a) The noise $G(t)$ with its expectation and variance are set to $(0, 25)$. (b) The system error $e(t)$ with noise when $m = 30$ and $k = 32$. (c) The comparison of estimated value and the accurate value of the 30 Walsh integrated messages with noise. (d) The comparisons of recovered value and original value in the 960 messages with noise.

Notations

$x(t)$: The state variable of the drive system
$y(t)$: The state variable of the response system
$e(t)$: The state variable of the response system
τ, τ_i: The time delays
a_i: System parameters
$\widehat{a}_i$: The estimated value of a_i
m: The number of system parameters
$\alpha, \beta, \gamma, \eta$: Constants
k: The order of Walsh function
$W_j(t)$: The jth Walsh sequence of k-orders Walsh function
$\widehat{b}_i$: The estimated value of $\sum_{j=1}^{k} a_{ij} W_j(t)$
l_b: The length of one bit of information
l_w: The length of each k-orders Walsh function's code element ($l_w = l_b/k$)
t_j: The sample time ($t_j = jl_w$)
H: The effective length of the carrier
Q: The quantity of total information (bits) carried by the system.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grant nos. 61472045 and 61573067), the National Key Research and Development Program (Grant no. 2016YFB0800602), the Beijing City Board of Education Science and Technology Key Project (Grant no. KZ201510015015), and the Beijing City Board of Education Science and Technology Project (Grant no. KM201510015009).

References

[1] J. Chen, Q. Liang, B. Zhang et al., "A new secure transmission for big data based on nested sampling and coprime sampling," in *The Proceedings of the Second International Conference on Communications, Signal Processing, and Systems*, pp. 733–741, Springer, 2014.

[2] J. Manyika, M. Chui, B. Brown et al., "Big data: the next frontier for innovation, comptetition, and productivity," *Analytics*, 2011.

[3] G. Kaddoum, E. Soujeri, and Y. Nijsure, "Design of a short reference noncoherent chaos-based communication systems," *IEEE Transactions on Communications*, vol. 64, no. 2, pp. 680–689, 2016.

[4] M. F. Hassan, "Synchronization of uncertain constrained hyperchaotic systems and chaos-based secure communications via a novel decomposed nonlinear stochastic estimator," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2183–2211, 2016.

[5] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.

[6] T. Heil, I. Fischer, W. Elsässer, J. Mulet, and C. R. Mirasso, "Chaos synchronization and spontaneous symmetry-breaking in symmetrically delay-coupled semiconductor lasers," *Physical Review Letters*, vol. 86, no. 5, pp. 795–798, 2001.

[7] S. Hayes, C. Grebogi, and E. Ott, "Communicating with chaos," *Physical Review Letters*, vol. 70, no. 20, pp. 3031–3034, 1993.

[8] Y.-N. Li, L. Chen, Z.-S. Cai, and X.-Z. Zhao, "Study on chaos synchronization in the Belousov-Zhabotinsky chemical system," *Chaos, Solitons & Fractals*, vol. 17, no. 4, pp. 699–707, 2003.

[9] C. Zhou and J. Kurths, "Dynamical weights and enhanced synchronization in adaptive complex networks," *Physical Review Letters*, vol. 96, no. 16, Article ID 164102, 2006.

[10] L. Li, H. Peng, X. Wang, and Y. Yang, "Comment on two papers of chaotic synchronization," *Physics Letters A*, vol. 333, no. 3-4, pp. 269–270, 2004.

[11] N. J. Corron and J. N. Blakely, "Chaos in optimal communication waveforms," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2180, pp. 134–139, 2015.

[12] J. Gleick and R. C. Hilborn, "Making a new science," *Physics Today*, vol. 41, no. 11, p. 79, 1987.

[13] G. Álvarez, F. Montoya, M. Romera, and G. Pastor, "Breaking two secure communication systems based on chaotic masking," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 10, pp. 505–506, 2004.

[14] V. Milanović and M. E. Zaghloul, "Improved masking algorithm for chaotic communications systems," *Electronics Letters*, vol. 32, no. 1, pp. 11–12, 1996.

[15] G. Kolumban, P. M. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos. II. Chaotic modulation and chaotic synchronization," *IEEE Transactions on Circuits & Systems I Fundamental Theory & Applications*, vol. 45, no. 11, pp. 1129–1140, 1998.

[16] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, "Synchronization of Lorenz-based chaotic circuits with applications to communications," *IEEE Transactions on Circuits and Systems II: Analog & Digital Signal Processing*, vol. 40, no. 10, pp. 626–633, 1993.

[17] T. Yang and L. O. Chua, "Secure communication via chaotic parameter modulation," *IEEE Transactions on Circuits & Systems I: Fundamental Theory & Applications*, vol. 43, no. 9, pp. 817–819, 1996.

[18] D. Huang, "Synchronization-based estimation of all parameters of chaotic systems from time series," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, Article ID 067201, 2004.

[19] F. Tang, "An adaptive synchronization strategy based on active control for demodulating message hidden in chaotic signals," *Chaos, Solitons & Fractals*, vol. 37, no. 4, pp. 1090–1096, 2008.

[20] X.-J. Wu, H. Wang, and H.-T. Lu, "Hyperchaotic secure communication via generalized function projective synchronization," *Nonlinear Analysis. Real World Applications*, vol. 12, no. 2, pp. 1288–1299, 2011.

[21] G. Mazzini, G. Setti, and R. Rovatti, "Chaotic complex spreading sequences for asynchronous DS-CDMA. I. System modeling and results," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 10, pp. 937–947, 1997.

[22] T. Yang and L. O. Chua, "Chaotic digital code-division multiple access (CDMA) communication systems," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 7, no. 12, pp. 2789–2805, 1997.

[23] H.-P. Ren, C. Bai, J. Liu, M. S. Baptista, and C. Grebogi, "Experimental validation of wireless communication with chaos," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, no. 8, Article ID 083117, 2016.

[24] G. M. Maggio, N. Rulkov, and L. Reggiani, "Pseudo-chaotic time hopping for UWB impulse radio," *IEEE Transactions on Circuits and Systems I: Fundamental Theory & Applications*, vol. 48, no. 12, pp. 1424–1435, 2001.

[25] V. Annovazzi-Lodi and G. Aromataris, "Privacy in two-laser and three-laser chaos communications," *IEEE Journal of Quantum Electronics*, vol. 51, no. 7, pp. 1–5, 2015.

[26] F. Kuwashima, T. Shirao, T. Kishibata et al., "High effective generation and detection of THz waves using a laser chaos and a super-focusing with metal V-grooved waveguides," in *Proceedings of the 40th International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz '15)*, Hong Kong, China, August 2015.

[27] O. I. Moskalenko, A. A. Koronovskii, and A. E. Hramov, "Generalized synchronization of chaos for secure communication: remarkable stability to noise," *Physics Letters, Section A: General, Atomic and Solid State Physics*, vol. 374, no. 29, pp. 2925–2931, 2010.

[28] D. Ghosh, "Nonlinear-observer–based synchronization scheme for multiparameter estimation," *Europhysics Letters*, vol. 84, no. 4, Article ID 40012, pp. 605–609, 2008.

[29] D. Ghosh and A. Roy Chowdhury, "Lag and anticipatory synchronization based parameter estimation scheme in modulated time-delayed systems," *Nonlinear Analysis: Real World Applications*, vol. 11, no. 4, pp. 3059–3065, 2010.

[30] D. Huang, G. Xing, and D. W. Wheeler, "Multiparameter estimation using only a chaotic time series and its applications," *Chaos. An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, pp. 471–516, 2007.

[31] F. Sun, L. Li, H. Peng, C. Wang, and Y. Yang, "Multiple information transmission using only one scalar chaotic time series," *The European Physical Journal B*, vol. 86, no. 2, article 39, 2013.

[32] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, vol. 85, no. 2, University of Illinois, Urbana University of Illinois Press, 1949.

[33] R. V. L. Hartley, *Transmission of Information*, The M.I.T.Pr. and John W, 1965.

[34] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[35] F. Sun, H. Peng, Q. Luo, L. Li, and Y. Yang, "Parameter identification and projective synchronization between different chaotic systems," *Chaos*, vol. 19, no. 2, p. 259, 2009.

[36] H. Peng, L. Li, Y. Yang, and F. Sun, "Conditions of parameter identification from time series," *Physical Review E: Statistical, Nonlinear, & Soft Matter Physics*, vol. 83, no. 3, part 2, pp. 989–1010, 2011.

[37] H. F. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer, 1972.

[38] H. Yin, X. Chen, H. Yue et al., "Experimental realization of long-haul chaotic optical secure communications," in *Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '15)*, pp. 2112–2116, Zhangjiajie, China, August 2015.