IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 1, 2016, accepted December 29, 2016, date of publication January 16, 2017, date of current version March 8, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2652486

Hierarchical Trust Level Evaluation for Pervasive Social Networking

JIAN SHEN[1], (Member, IEEE), TIANQI ZHOU[2], CHIN-FENG LAI[3], (Senior Member, IEEE), JIGUO LI[4], AND XIONG LI[5]

[1]Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan
[4]College of Computer and Information Engineering, Hohai University, Nanjing 210044, China
[5]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

Corresponding author: C.-F. Lai (cinfon@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672295, Grant 61300237, Grant U1536206, Grant U1405254, Grant 61232016, and Grant 61402234; in part by the National Basic Research Program 973 under Grant 2011CB311808; in part by the Natural Science Foundation of Jiangsu province under Grant BK2012461; in part by the research fund from Jiangsu Technology and Engineering Center of Meteorological Sensor Network in NUIST under Grant KDXG1301; in part by the research fund from Jiangsu Engineering Center of Network Monitoring in NUIST under Grant KJR1302; in part by the research fund from the Nanjing University of Information Science and Technology under Grant S8113003001; in part by the 2013 Nanjing Project of Science and Technology Activities for Returning from Overseas; in part by the 2015 Project of six personnel in Jiangsu Province under Grant R2015L06, in part by the CICAEET fund, and in part by the PAPD fund.

ABSTRACT Pervasive social networking (PSN) is a fundamental infrastructure in social networking that has played an important role in not only the Internet but also mobile domains. A practical and accurate evaluation system is required to ensure the further development of PSN. Secure and efficient communication is also an essential issue in PSN to increase its adoption in daily life. In this paper, we discuss the establishment of a hierarchical evaluation system to support secure and trustworthy PSN with multiple and variable nodes. The proposed hierarchical evaluation system is essentially based on a special symmetric balanced incomplete block design: the (7, 3, 1)-design and the tree structure. Together, they constitute a multilevel system that supports both our hierarchical trust level (HTL) evaluation system and key agreement scheme. The former solves the problem of trust evaluation in PSN, and the latter guarantees the secure communication of trusted nodes. Note that both security and performance analyses show that the proposed HTL evaluation system can support extensive adoption of efficient and secure PSN.

INDEX TERMS Pervasive social networking (PSN), trust evaluation, secure communication, hierarchical trust level (HTL).

I. INTRODUCTION

With the popularity of smart devices (e.g., mobile phones), pervasive social networking (PSN) is no longer merely a complement to Internet social networking but is becoming its main trend [1]. A social network is a social structure made up of a set of social actors (e.g., individuals or organizations), sets of dyadic ties, and other social interactions between actors [2]. Essentially, PSN is a universal and pervasive social networking that supports various types of instant social networking. It is necessary that PSN can be achieved and accessed at any time and location.

In recent years, social network sites (e.g., Myspace, Facebook, Bebo, Orkut, LinkedIn) have rapidly gained popularity among all types of people. PSN offers people a platform to chat, make friends and play games, thereby providing many opportunities to interact with other people and learn what is happening in the world. For instance, one can use a computer or cell phone to log on at home, in a restaurant, or even in the subway in a convenient way. In addition, PSN can be applied to wearable health care services [3], [4] to offer instant and convenient communication. There are signs that PSN will eventually become widespread.

Groups of people can share interests and interact in a variety of ways via PSN. These include file-sharing, chatting, messaging, and exchanging photos and videos. It is obvious that trust plays a very important role in communication between strangers. Without a credible and efficient evaluation system it is difficult to construct a complete PSN.

Guaranteeing communication security in instant messaging is also a key problem in the construction of a social network because many users' private information is contained in PSN [5]; the difficulty of protecting users' privacy has hindered its further popularization. Unless the issues of credible evaluation and the security of communication are addressed, many potential customers will be reluctant to use it.

A. CHALLENGES IN PSN

The following are the two major issues of PSN that affect its development and popularization.

1. How to protect the communications among trusted nodes.

2. How to establish an effective evaluation system among nodes [6].

In [7], social networking was studied by Emre *et al.* based on the mobile ad hoc network (MANET) [8]. Trust and reputation, however, have not been considered in the literature. In traditional social networking, trust is an evaluation that can be derived from direct or indirect knowledge. Based on the trust level, one can assess the advisable level of belief in an entity. Note that under this mechanism, trust is stored in the entity, which provides the opportunity to deceive customers by arbitrarily expanding their trust to pursue illegitimate interests. Even if no particular benefit is sought, the best survival strategy when strangers meet seems to be to cheat [9]. This is especially useful when a person does not expect to see the other party again. In [10], a trust management system is proposed that combines a local trust (LT) level evaluation and a general trust (GT) level evaluation to encourage good behavior in PSN, but the problems of protecting communications among trustworthy nodes have not been solved [10].

To avoid eavesdropping from malicious nodes and protect data transmission and processing, it is essential to protect the data communications in PSN [11]. In the current literature, the most extensive survey of privacy concerns considers the user's personal data and location [12]–[14]. To ensure the user's privacy, commonly used solutions include data encryption and key distribution. In particular, in [13], a key distribution scheme is proposed, and in [15], a server is introduced to issue anonymous identities. Recently, many social networking applications have attempted to address the privacy concerns. For example, Sadeh *et al.* and Miluzzo *et al.* discuss data privacy issues in PeopleFinder and CenceMe, respectively [16], [17].

Therefore, an effective trust level evaluation system and a secure scheme are required to widen the adoption of PSN.

B. OUR CONTRIBUTION

It is of paramount importance to establish an accurate and practical trust level evaluation system for PSN. The contributions of this paper are listed as follows.

1) WE PROPOSED A HIERARCHICAL SYSTEM TO REALIZE NOVEL TRUST MANAGEMENT

In our scheme, the trust level of the nodes is generated according to the LT or GT, which are defined in Section II. It is worth noting that based on the (7, 3, 1)-design, the communication cost to evaluate the trust level of a node in a group is only $O(n\sqrt{n})$, where n is the number of nodes in the group.

2) WE ADDRESSED THE PROBLEM OF SECURE COMMUNICATION AMONG TRUSTWORTHY NODES

In spite of generating the trust level of every node, our scheme can also derive a common session key to ensure the secure communication of nodes in PSN. Moreover, an authentication service is supported based on the shared key.

3) WE PROPOSED A RELIABLE EVALUATION AND SECURE COMMUNICATION SYSTEM FOR PSN

The combination of the (7, 3, 1)-design and the tree structure supports an efficient trust evaluation mechanism of multiple nodes in PSN. Based on this system, the trust level of a node is evaluated by all of its group members or a trusted server (TS). Additionally, a secure session key is generated to ensure secure communication among the group nodes.

4) WE PROPOSED A SECURE SCHEME TO SUPPORT AUTHENTICATION SERVICES AND ACCESS CONTROL

Authentication is the act of confirming the truth of an attribute of a single piece of data claimed true by an entity [18]–[21]. In simple terms, authentication is to identify the user's identity through a certain means. There are many methods of authentication, including authentication based on a shared key, authentication based on biological features [22], [23] and authentication based on a public key encryption algorithm. Our key agreement scheme allows each user to obtain a shared secret key, and users with the shared key can then realize the identity authentication and access control [24].

C. ORGANIZATION

The rest of this paper is organized as follows. Section II describes some related works. Section III presents some preliminaries and provides some definitions. In Section IV, a hierarchical trust level (HTL) evaluation system is proposed, following by a performance analysis in Section V. Section VI presents further discussion on security in PSN and the security analysis, and conclusions are drawn in Section VII.

II. RELATED WORKS

In [11] and [1], the trust level of nodes in PSN is defined, and a trust management framework is proposed. Based on the framework in [11], secure PSN with two-dimensional levels is realized. Many works exist on trust management and secure PSN [25]–[28].

Definition 1: GT is the evaluation indicator of a node trust level according to the information collected by the TS from all the nodes.

Definition 2: LT is the evaluation indicator of a node trust level according to the information collected by the PSN nodes.

Node \ Node	1	2	3	4	5	6	7
1	/	4	2	0	0	6	2
2	3	/	2	1	5	7	4
3	4	5	/	3	0	3	5
4	2	5	3	/	1	4	5
5	1	2	1	3	/	5	3
6	5	1	5	2	2	/	6
7	6	3	6	5	3	2	/

FIGURE 1. Trust level matrix *TL*.

The GT and LT are the trust levels of a PSN node and can be divided into discrete levels. In this paper, GT_i presents the i^{th} level of the GT and LT_j denotes the j^{th} level of the LT, $i \in [0, MAX_{GT}]$ and $j \in [0, MAX_{LT}]$, where MAX_{GT} and MAX_{LT} are the maximum levels of the GT and LT, respectively. In this paper, we define the maximum levels of the GT and LT to be 6, namely, $MAX_{GT} = MAX_{LT} = 6$. Thus, GT_i and LT_j are integers between 0 and 6.

The GT evaluation requires the TS to collect all of the user's information, which will introduce huge overhead to the server. Additionally, relying solely on the information gathered by the TS increases the possibility of network congestion and may also omit the information of some nodes. In practice, it is likely to be very misleading to rely solely on the information collected by the TS. The causes of this problem are varied. The first is the integrity of the information collection. In instant communication, it is difficult to collect the whole information of each node, so the TS cannot make accurate evaluations. Second, an evaluation made according to the behavior of a node that is observed only once or twice is likely to be incorrect. Additionally, in such a system, the TS will be the key point of the evaluation system, and if the TS fails, the entire evaluation system becomes ineffective. Finally, relying solely on a third party for the evaluation may introduce certain security risks (e.g., attack on the TS or the TS pursuing illegitimate interests).

The LT level of a node is based on the evaluation of all PSN nodes. This system requires each node to obtain an evaluation of the rest of the nodes, which introduces much communication complexity. Moreover, in PSN with multiple nodes, many nodes are not familiar with the other nodes, which can easily lead to a false or invalid evaluation.

A. TRADITIONAL TRUST LEVEL EVALUATION

In PSN with 7 nodes, the common trust level evaluation according to the GT and LT is described in detail as follows. The GT requires the TS to accumulate all of the trust levels of the nodes, whereas the LT requires all of the nodes to acquire its own trust level through all of the remaining nodes. In Fig. 1, a square matrix of order 7 is used to represent the node's trust level, which is denoted as *TL*. Specifically, $TL[i][j]$ represents the trust evaluation on node i from node j. The i^{th} row is the trust evaluation on node i from all of the remaining nodes, and the i^{th} column is the trust evaluation from node i on all of the remaining nodes.

For example, $TL[3][2] = 5$, which represents that the trust level on node 3 from node 2 is 5. $TL[4] = \{2, 5, 3, 1, 4, 5\}$ represents that the trust levels on node 4 from nodes 1, 2, 3, 5, 6, 7 are 2, 5, 3, 1, 4, 5, respectively, and the 4^{th} column indicates that the trust levels to nodes 1, 2, 3, 5, 6, 7 from node 4 are 0, 1, 3, 3, 2, 5, respectively. Here, we believe that each node is ineffective in its own evaluation, so the value of the diagonal of the square is invalid, and it is marked with "/" in Fig. 1.

In the GT-based evaluation system, the TS needs to accumulate the trust level of each row. This evaluation system is very intuitive and easy to implement. However, as we mentioned above, it is an ideal model in PSN and increases the security risks of the system.

In the LT-based evaluation system, each node can acquire the trust evaluation on it from all of the remaining nodes. The detailed process is shown in Fig. 2; here, every node collects the trust levels from the other 6 nodes and calculates the trust level according to Eq. 1, where TL_i represents the trust level of node i and n is the number of nodes in PSN.

$$TL_i = \left\lceil \frac{\sum_{x=1}^{n} TL[x][i]}{n-1} \right\rceil, \quad (x \neq i). \tag{1}$$

Based on the collection process of the trust level in Fig. 2 and the calculation in Eq. 1, the traditional trust level evaluation on PSN with 7 nodes has the following results.

$$TL_1 = \left\lceil \frac{4+2+0+0+6+2}{7-1} \right\rceil = 3$$
$$TL_2 = \left\lceil \frac{3+2+1+5+7+4}{7-1} \right\rceil = 4$$
$$TL_3 = \left\lceil \frac{4+5+3+0+3+5}{7-1} \right\rceil = 4$$
$$TL_4 = \left\lceil \frac{2+5+3+1+4+5}{7-1} \right\rceil = 4$$
$$TL_5 = \left\lceil \frac{1+2+1+3+5+3}{7-1} \right\rceil = 3$$
$$TL_6 = \left\lceil \frac{5+1+5+2+2+6}{7-1} \right\rceil = 4$$
$$TL_7 = \left\lceil \frac{6+3+6+5+3+2}{7-1} \right\rceil = 5$$

Although this evaluation system can reduce the security risks of the GT and the burden of the TS, it needs more communication overhead. In particular, when the number of nodes increases, the communication overhead experiences quadratic growth. Further, using the LT evaluation system in a large number of nodes in PSN to derive the trust level is not accurate. The root cause of this problem is that we

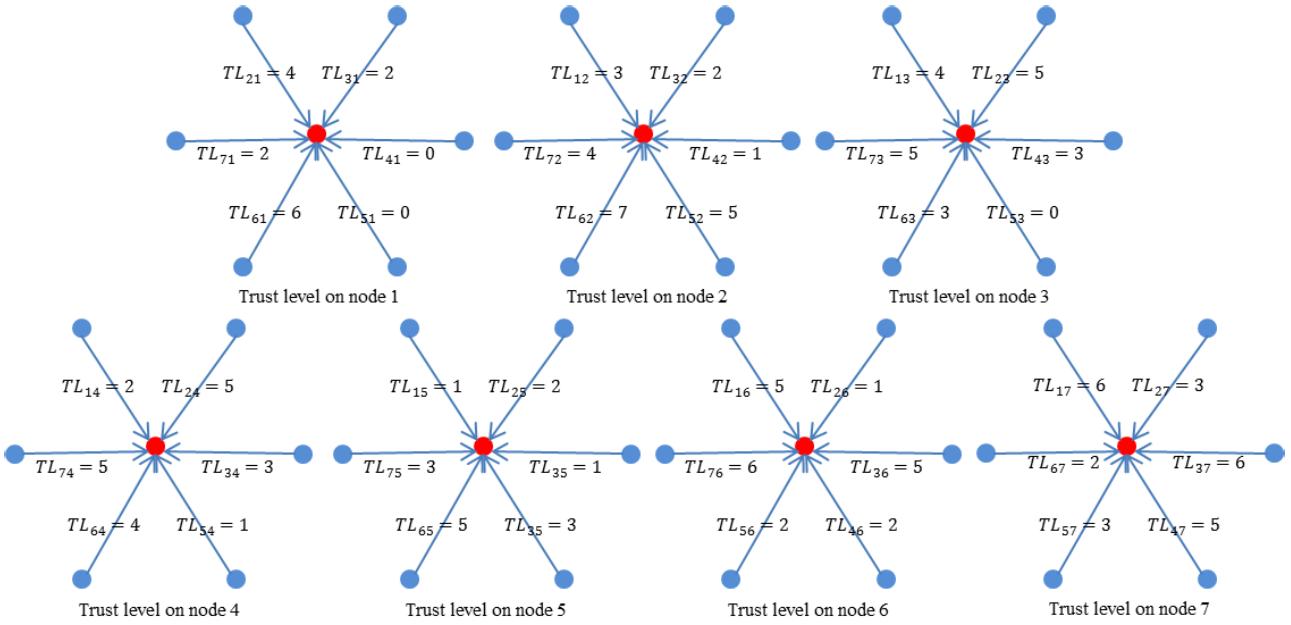


FIGURE 2. Traditional LT-based evaluation system.

usually tend to be in a circle or a few circles in social networks and will naturally be more familiar with a certain area and unfamiliar with other areas. Thus, using LT in PSN with a large number of nodes is unrealistic.

Therefore, in this paper, we propose a HTL evaluation system that is based on the (7, 3, 1)-design and tree structure.

III. PRELIMINARIES AND DEFINITIONS

A. BILINEAR MAPS

Definition 3: Let G_1 and G_2 be two groups of order q for some large prime q. Let $\mathcal{G}$ be the generator of G_1 and G_2. A modified Weil pairing is a map $\hat{e} : G_1 \times G_2 \rightarrow G_2$, which has the following properties [29].

1. Bilinear: For any $\mathcal{P}, \mathcal{Q} \in G_1$ and $a, b \in Z$, we have $\hat{e}(a\mathcal{P}, b\mathcal{Q}) = \hat{e}(\mathcal{P}, \mathcal{Q})^{ab}$.

2. Non-degenerate: If $\mathcal{G}$ is a generator of G_1, then $\hat{e}(\mathcal{G}, \mathcal{G}) \in F^*_{p^2}$ is a generator of G_2 . In other words, $\hat{e}(\mathcal{G}, \mathcal{G}) \neq 1$.

3. Non-commutative: For any $\mathcal{P}, \mathcal{Q} \in G_1$, $\hat{e}(\mathcal{P}, \mathcal{Q}) = \hat{e}(\mathcal{Q}, \mathcal{P})$.

4. Computable: Given $\mathcal{P}, \mathcal{Q} \in G_1$, there exists an efficient algorithm to compute $e(\mathcal{P}, \mathcal{Q})$

5. For any $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2 \in G_1$, $\hat{e}(\mathcal{P}_1 + \mathcal{P}_2, \mathcal{Q}_1) = \hat{e}(\mathcal{P}_1, \mathcal{Q}_1) \cdot \hat{e}(\mathcal{P}_2, \mathcal{Q}_1)$. Similarly, $\hat{e}(\mathcal{P}_1, \mathcal{Q}_1 + \mathcal{Q}_2) = \hat{e}(\mathcal{P}_1, \mathcal{Q}_1) \cdot \hat{e}(\mathcal{P}_1, \mathcal{Q}_2)$.

B. BILINEAR DIFFIE-HELLMAN ASSUMPTION (BDH)

Given $\langle \mathcal{G}, a\mathcal{G}, b\mathcal{G}, c\mathcal{G} \rangle$ for some $a, b, c \in Z^*_q$, compute $W = \hat{e}(\mathcal{G}, \mathcal{G})^{abc} \in G_2$, where $\mathcal{G}$ is a generator of G_1. An algorithm $\mathcal{A}$ has an advantage ε in solving BDH in $\langle G_1, G_2, \hat{e} \rangle$ if

$$\Pr\left[\mathcal{A}(\mathcal{G}, a\mathcal{G}, b\mathcal{G}, c\mathcal{G}) = \hat{e}(\mathcal{G}, \mathcal{G})^{abc}\right] \geq \varepsilon.$$

The probability is over the random choice of $a, b, c \in Z^*_q$, the random choice of $\mathcal{G} \in G_1$, and the random bits of $\mathcal{A}$ [30].

C. IDENTITY-BASED ENCRYPTION

An identity-based encryption scheme (IBE) can be described as follows [31], [32]:

1) SETUP

Input a security parameter k and return *params* (system parameters), which includes a description of a finite message space M and a description of a finite cipher text space C. It also returns a master-key which will be known by the private key generator (PKG).

2) EXTRACT

Input *params*, master-key and an arbitrary $ID \in \{0, 1\}^*$, which is an arbitrary string that will be used as a public key. Then, it returns a private key d which is the corresponding private key of ID. This step is to extract a private key from the given public key.

3) ENCRYPT

Takes *params*, ID and $m \in M$ as inputs. It returns a cipher text $c \in C$.

4) DECRYPT

Takes *params*, $c \in C$, and a private key d as inputs. It returns $m \in M$.

An IBE must satisfy the standard consistency constraint, that is, when d is the private key generated by Step 3 Encrypt,

then

$$\forall m \in M : Decrypt(params, C, d) = M$$

where $C = Encrypt(params, ID, M)$ and ID is the corresponding public key of d.

D. BLOCK DESIGN

A block design is a set together with a family of subsets whose members are chosen to satisfy some set of properties that are deemed useful for a particular application. A balanced incomplete block design (BIBD) is defined below [33].

Definition 4: Let $V = \{0, 1, 2 \ldots v-1\}$ be a set of v elements and $B = \{B_0, B_1, B_2 \ldots B_{b-1}\}$ be a set of b blocks, where B_i is a subset of V and $|B_i| = k$. For a finite incidence structure $\sigma = \{V, B\}$, if σ satisfies the following conditions, then it is a BIBD, which is called a (b, v, r, k, λ)-design.

1) Each element appears in exactly r of the b blocks.
2) Every two elements appear simultaneously in exactly λ of the b blocks.
3) $k < v$, so that no block contains all of the elements of set V.
4) $b \geq v$. The case of equality is called a symmetric design.

For a (b, v, r, k, λ)-design, if it satisfies $k = r$ and $b = v$, then it is a symmetric balanced incomplete block design(SBIBD), which is called a (v, k, λ)-design. A concrete example of the SBIBD is shown in Fig. 3.

IV. HIERARCHICAL TRUST LEVEL EVALUATION SYSTEM

In our hierarchic trust level evaluation architecture, the (7, 3, 1)-design is the basic structure. According to Definition 4, the (7, 3, 1)-design is a SBIBD, where $b = v = 7$, $k = r = 3$, $\lambda = 1$. A (7, 3, 1)-design structure is illustrated in Fig. 3. It is shown in Fig. 3 that $V = \{1, 2 \ldots 7\}$ is the set of 7 elements and $B = \{B_1, B_2, B_3 \ldots B_7\}$ is the set of 7 blocks. Moreover, each block contains 3 elements, each element appears in exactly 3 of the 7 blocks (e.g., $B_1 = \{1, 2, 4\}$ and $1 \in B_1, B_5, B_7$), and every two elements appear simultaneously in exactly one of the 7 blocks (e.g., {1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {1, 7}).

A. TRUST LEVEL EVALUATION BASED ON THE (7, 3, 1)-DESIGN

This evaluation system requires two steps to derive a trust level for every node. Each step is on the basis of the (7, 3, 1)-design structure, which is shown in Fig. 3.

1) STEP 1

Node i in block B_i collects the trust value on it from the other 2 nodes in block B_i. Note here that the following processes is the key point of using the (7, 3, 1)-design. In addition to obtaining the evaluation of its own trust level, node i needs to evaluate the trust level on the remaining two nodes, and it is more important that the remaining two nodes obtain the evaluations on each other. The previous two trust evaluations on node i will contribute to the final trust level of node i, and the later four trust evaluations will be stored in node i.

B \ V	1	2	3	4	5	6	7
1	1	2		4			
2		2	3		5		
3			3	4		6	
4				4	5	7	
5	1				5	6	
6		2				6	7
7	1		3				7

FIGURE 3. (7, 3, 1)-design structure.

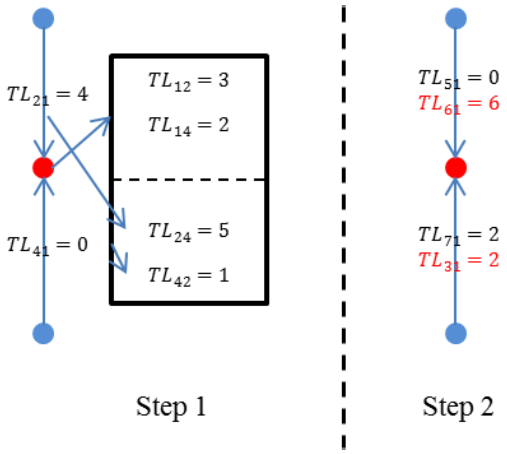


FIGURE 4. Trust levels collected by node 1.

For example, since $B_1 = \{1, 2, 4\}$, in Step 1, node 1 collects the trust level on it from nodes 2 and 4, which contribute to its final trust level. Then, the trust evaluation of node 1 to nodes 2, 4 and the trust evaluations of node 2 to node 4 and node 4 to node 2 are stored in node 1.

2) STEP 2

Node i collects a trust evaluation from node j if node i is contained in the j^{th} block B_j in the (7, 3, 1)-design. After that, node i can collect all of the trust evaluations on itself.

For example, in Step 2, node 1 collects the trust level on it from nodes 5 and 7. It is worth noting that node 1 can not only collect the trust evaluations from nodes 5 and 7 but also from nodes 6 and 3. The detailed trust level collection process of node 1 in Step 1 and Step 2 is shown in Fig. 4

Finally, based on the (7, 3, 1)-design, every node can collect all of the trust evaluations from the remaining nodes. The detailed process of our trust evaluation system is illustrated in Table 1.

In Table 1, in Step 1, each node collects the trust evaluations from the corresponding nodes according to the structure of the (7, 3, 1)-design and stores the relevant evaluation that can be accessed.

TABLE 1. (7, 3, 1)-design trust evaluation system.

Node	Step 1	Step 1
$Node1$	$TL_{21}=4; TL_{41}=0$	$TL_{51}=0; TL_{61}=6;$ $TL_{71}=2; TL_{31}=2$
$Node2$	$TL_{32}=2; TL_{52}=5$	$TL_{12}=3; TL_{62}=7;$ $TL_{42}=1; TL_{72}=4$
$Node3$	$TL_{43}=3; TL_{63}=3$	$TL_{23}=5; TL_{73}=5;$ $TL_{53}=0; TL_{13}=4$
$Node4$	$TL_{54}=1; TL_{74}=5$	$TL_{14}=2; TL_{34}=3;$ $TL_{24}=5; TL_{64}=4$
$Node5$	$TL_{15}=1; TL_{65}=5$	$TL_{25}=2; TL_{45}=3;$ $TL_{35}=1; TL_{75}=3$
$Node6$	$TL_{26}=1; TL_{76}=6$	$TL_{36}=5; TL_{56}=2;$ $TL_{46}=2; TL_{16}=5$
$Node7$	$TL_{17}=6; TL_{37}=6$	$TL_{47}=5; TL_{67}=2;$ $TL_{57}=3; TL_{27}=3$

- *Node*1 collects $TL_{21}=4; TL_{41}=0$ from *Node*2 and *Node*4, respectively, and stores $TL_{24}=5; TL_{42}=1$.
- *Node*2 collects $TL_{32}=2; TL_{52}=5$ from *Node*3 and *Node*5, respectively, and stores $TL_{35}=1; TL_{53}=0$.
- *Node*3 collects $TL_{43}=3; TL_{63}=3$ from *Node*4 and *Node*6, respectively, and stores $TL_{46}=2; TL_{64}=4$.
- *Node*4 collects $TL_{54}=1; TL_{74}=5$ from *Node*5 and *Node*7, respectively, and stores $TL_{57}=3; TL_{75}=3$.
- *Node*5 collects $TL_{15}=1; TL_{65}=5$ from *Node*1 and *Node*6, respectively, and stores $TL_{16}=5; TL_{61}=6$.
- *Node*6 collects $TL_{26}=1; TL_{76}=6$ from *Node*2 and *Node*7, respectively, and stores $TL_{27}=3; TL_{72}=4$.
- *Node*7 collects $TL_{17}=6; TL_{37}=6$ from *Node*1 and *Node*3, respectively, and stores $TL_{13}=4; TL_{31}=2$.

In Step 2, each node only needs to collect the trust evaluation from 2 nodes to collect all of the trust evaluations on itself.

- *Node*1 collects $TL_{51}=0; TL_{71}=1$ from *Node*5 and *Node*7, respectively. Note here that *Node*1 can acquire $TL_{61}=6; TL_{31}=2$, which are stored in *Node*5 and *Node*7 in Step 1, respectively.
- *Node*2 collects $TL_{12}=3; TL_{62}=7$ from *Node*1 and *Node*6, respectively. Note here that *Node*2 can acquire $TL_{42}=1; TL_{72}=4$, which are stored in *Node*1 and *Node*6 in Step 1, respectively.
- *Node*3 collects $TL_{23}=5; TL_{73}=5$ from *Node*2 and *Node*7, respectively. Note here that *Node*3 can acquire $TL_{53}=0; TL_{13}=4$, which are stored in *Node*2 and *Node*7 in Step 1, respectively.
- *Node*4 collects $TL_{14}=2; TL_{34}=3$ from *Node*1 and *Node*3, respectively. Note here that *Node*4 can acquire $TL_{24}=5; TL_{64}=4$, which are stored in *Node*1 and *Node*3 in Step 1, respectively.
- *Node*5 collects $TL_{25}=2; TL_{45}=3$ from *Node*2 and *Node*4, respectively. Note here that *Node*1 can acquire $TL_{35}=1; TL_{75}=3$, which are stored in *Node*2 and *Node*4 in Step 1, respectively.
- *Node*6 collects $TL_{36}=5; TL_{56}=2$ from *Node*3 and *Node*5, respectively. Note here that *Node*6 can acquire $TL_{46}=2; TL_{16}=5$, which are stored in *Node*3 and *Node*6 in Step 1, respectively.

Algorithm 1 Grouping Algorithm

while $n > 7$ **do**
  if $n\%7 == 0$ **then**
    $n = n/7$;
    $p_1 = p_2 = \ldots = p_7 = n$;
    Grouping(p_1),Grouping(p_2),Grouping(p_3),
    Grouping(p_4),Grouping(p_5),Grouping(p_6),
    Grouping(p_7);
  else
    $n = n + (7 - n\%7)$;
    $r = 7 - n\%7$;
    $n = n/7$;
    $p_1 = p_2 = \ldots = p_7 = n$;
    Grouping(p_1),Grouping(p_2),Grouping(p_3),
    Grouping(p_4),Grouping(p_5),Grouping(p_6),
    Grouping(p_7);
  end if
end while
$leaf_{number} = n$;

- *Node*7 collects $TL_{47}=5; TL_{67}=2$ from *Node*4 and *Node*6, respectively. Note here that *Node*7 can acquire $TL_{57}=3; TL_{27}=3$, which are stored in *Node*4 and *Node*6 in Step 1, respectively.

It is worth noting that compared with the common trust level evaluation system, the communication cost of our trust level evaluation system based on the (7, 3, 1)-design is diminished. Specifically, considering the example given above, in Fig. 2, each node evaluates all of the remaining 6 nodes, so the communication cost is $6 \times 7 = 42$. However, in our trust evaluation system, each node only needs to contact with the remaining 4 nodes, which results in a low communication overhead of $4 \times 7 = 28$.

Even if the number of nodes is small, the gap between the evaluation systems can be directly found. Therefore, we believe that the use of the (7, 3, 1)-design structure will greatly reduce the communication overhead in the trust level evaluation system.

In the next subsection, the (7, 3, 1)-design structure is combined with the tree structure to achieve a multiple-node HTL evaluation system.

B. HIERARCHICAL TRUST LEVEL EVALUATION

The number of nodes in the real PSN will not be fixed to 7, so we need a hierarchical evaluation system. In our HTL evaluation system, nodes will be grouped in sets of 7. Moreover, the GT and LT evaluation will be combined together to achieve an effective HTL evaluation system. GT(i) represents the common trust evaluation on node *i* from TS, while LT(I) represents the trust evaluation based on the (7, 3, 1)-design, where *I* is a set of 7 nodes. Each node in set *I* has a better understanding of the rest of the nodes.

Algorithm 1 is a recursive algorithm to construct tree structure based on any number of nodes in PSN. In Algorithm 1,

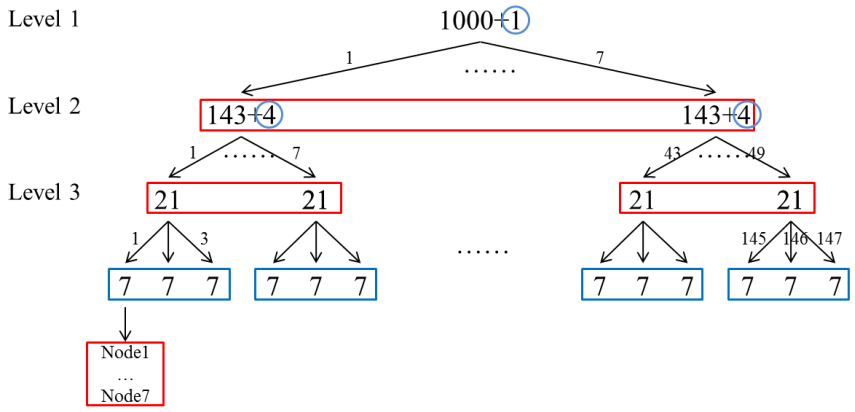


FIGURE 5. Hierarchical trust level evaluation.

n is the number of nodes in PSN, and $p_1, p_2, p_3, p_4, p_5, p_6, p_7$ are the numbers of PSN nodes contained in the child nodes. To ensure that the $(7, 3, 1)$-design can be used in multiple nodes in PSN, each recursion needs to make sure that the number of nodes is exactly divisible by 7. Otherwise, some nodes will be added by the TS to meet the limitation. The evaluation on the added nodes is given by the TS.

The tree constructed by Algorithm 1 has the following properties.

1. In addition to the last level, each node of the tree has 7 branches.
2. The level of the tree is $\lceil \log_7 n \rceil - 1$.
3. In addition to the leaf nodes, the number of nodes in the i^{th} level is 7^{i-1}.
4. The number of branches of each node in the last level is $leaf_{number}$, which is equal to $\left\lceil n \Big/ 7^{\lceil \log_7 n \rceil - 1} \right\rceil$.
5. In the last levels, each node has $leaf_{number}$ branches, and the other nodes in the tree have 7 branches.
6. The number of leaf nodes of the tree is $leaf_{number} \times 7^{\lceil \log_7 n \rceil - 2}$.

Figure 5 is the tree structure with 1000 PSN nodes. It is clear from the Fig. 5 that the level of the tree is $\lceil \log_7 1000 \rceil - 1 = 3$. In addition to the nodes in the last level, each node has 7 branches, and the number of nodes in the i^{th} level is 7^{i-1}. For example, in level 2, each node of the level has 7 branches, and the number of nodes in the level is $7^1 = 7$. The number of leaf nodes in each node of the second to last level is $\left\lceil 1000 \Big/ 7^{\lceil \log_7 1000 \rceil - 1} \right\rceil = 3$, and each of them contains 7 PSN nodes. The number of leaf nodes of the tree is $leaf_{number} \times 7^{\lceil \log_7 1000 \rceil - 2} = 3 \times 7^2 = 147$.

Our HTL evaluation system combines GT and the $(7, 3, 1)$-design LT. The GT evaluation is performed by the TS. The TS takes responsibility for issuing trust levels for the nodes, which cannot preform the $(7, 3, 1)$-design LT due to the structural constraints of the $(7, 3, 1)$-design. These nodes are marked with blue rectangles in Fig. 5. Similarly, in Fig. 5, the nodes marked with blue circles cannot preform the $(7, 3, 1)$-design LT. Therefore, these two types of nodes will obtain the GT from the TS.

After that, the nodes marked with red rectangles will preform the $(7, 3, 1)$-design based TL evaluation, which is described in detail in the previous section.

V. PERFORMANCE ANALYSIS

In this section, we will analyze the performance of the proposed hierarchical system in detail. According to the discussion in Section IV, the communication overhead required for the basic structure of the hierarchical evaluation system is 28. Then, based on the properties of the tree in the HTL evaluation system, we can derive the communication overhead of the PSN with n nodes as Eq.2.

$$28 \times (7^1 + 7^2 + 7^3 + \ldots + 7^{\lceil \log_7 n \rceil - 2} + \left\lceil n \Big/ 7^{\lceil \log_7 n \rceil - 1} \right\rceil \times 7^{\lceil \log_7 n \rceil - 2}). \quad (2)$$

For example, when the number of nodes is 100, the communication overhead is

$$comm\_HTL = 28 \times (7 + \left\lceil 100 \Big/ 7^2 \right\rceil \times 7^{\lceil \log_7 100 \rceil - 2}) = 28 \times (7 + 3 \times 7) = 784,$$

while the communication overhead of the common LT is

$$comm\_LT = 100 \times 99 = 9900.$$

Let μ_n represent the savings rate of the communication overhead when the number of nodes in the PSN is n, which is calculated as Eq. (3).

$$\mu_n = \frac{comm\_LT - comm\_HTL}{comm\_LT} \times 100\%. \quad (3)$$

According to Eq. (3), the savings rate of communication overhead when the number of nodes in the PSN is 100 is

$$\mu_{100} = \frac{9900 - 784}{9900} \times 100\% = 92.08\%.$$

In Table 2, a comparison of LT and HTL is illustrated. Compared with the common TL, the communication overhead is greatly reduced in the proposed HTL evaluation system. In particular, the greater the number of nodes in the

TABLE 2. Comparison of LT and HTL.

n	100	200	300	400	500	600	700	800	900	1000
LT	9900	3980	8970	159600	249500	359400	489300	639200	809100	999000
HTL	784	1176	1568	4312	4312	4312	5684	5684	5684	5684
μ	92.08%	70.55%	82.51%	97.29%	98.27%	98.8%	98.83%	99.11%	99.29%	99.43%

PSN is, the more obvious the advantage of this evaluation system is.

How to manage the nodes that are added by the TS and the node that cannot be evaluated based on the (7, 3, 1)-design structure is an open problem in the system.

VI. FURTHER DISCUSSION ON SECURE PSN

A. KEY AGREEMENT SCHEME BASED ON THE (7, 3, 1)-DESIGN

The two essential parts of PSN are the trust evaluation system and the secure communication among the nodes. In the previous section, we constructed a multi-level credit evaluation system, which is essentially based on a tree structure and the (7, 3, 1)-design of the SBIBD. In this section, based on the same structure, a secure key agreement scheme is proposed to support secure communication among the nodes in the PSN. In cryptography, a key agreement protocol is a protocol whereby two or more parties can agree on a key in such a way that both influence the outcome. The situation where two or more parties share a secret key is often called conference keying, and the shared secret key is often called a conference key. By employing a key agreement protocol, the conferees can securely send and receive messages from each other by using a common conference key that they agree upon in advance. A protocol that is useful in practice also does not reveal to any eavesdropping party what key has been agreed upon.

In the proposed scheme, the TS takes responsibility for generating some system parameters and distributing the private key for nodes in the (7, 3, 1)-design. First, the TS publishes $\{p, q, G_1, G_2, \mathcal{G}, \hat{e}, P_{pub}, H_1, H_2\}$, but keeps his private key $s \in Z_q^*$ secret. Here, p and q are two prime numbers, while $\mathcal{G}$, G_1, G_2 and $\hat{e}$ are the parameters of the Weil pairing, which are defined in Definition 3. In addition, H_1 and H_2 are two hash functions, which map arbitrary lengths to a nonzero point of G_1 and nonzero integer, respectively. In our identity-based key agreement protocol, the public key and private key of a node i are mapped as $H_1(ID_i)$ and $\mathcal{S}_i = sH_1(ID_i)$, respectively. Moreover, to provide authentication, based on the RSA cryptographic algorithm [34], the TS selects a public key e_i and a private key d_i for each node and distributes (e_i, n) to all the nodes, where n is the product of two large prime numbers. Then, node i computes $Y_i = H_2(ID_i)$, $X_i = (Y_i)^{d_i}$ and keeps (d_i, X_i) secret.

To derive a common session key, each node chooses a random number r_i and calculates the partial private key $\mathcal{M}_i = \hat{e}(\mathcal{G}, e_i r_i \mathcal{S}_i)$, which contributes to generating a common session key among all nodes. Notably, key agreement and trust evaluation are synchronous; that is, during the collection of the trust evaluation, each node also collects the corresponding partial private keys. Finally, every node obtains the common session key $\mathcal{K}$. Taking node 1 as an example, the common session key is calculated as Eq. 4.

$$\begin{aligned}\mathcal{K} &= \mathcal{M}_1 \cdot \mathcal{M}_2 \cdot \mathcal{M}_4 \cdot \mathcal{M}_5 \cdot \mathcal{M}_6 \cdot \mathcal{M}_3 \cdot \mathcal{M}_7 \\ &= \hat{e}(\mathcal{G}, e_1 r_1 S_1) \cdot \hat{e}(\mathcal{G}, e_2 r_2 \mathcal{S}_2 + e_4 r_4 \mathcal{S}_4) \\ &\quad \times \hat{e}(\mathcal{G}, e_5 r_5 \mathcal{S}_5 + e_6 r_6 \mathcal{S}_6) \cdot \hat{e}(\mathcal{G}, e_3 r_3 S_3 + e_7 r_7 S_7) \\ &= \hat{e}(\mathcal{G}, \sum_{i=1}^{7} e_i r_i \mathcal{S}_i). \end{aligned} \tag{4}$$

Similar to node 1, each node in the (7, 3, 1)-design can obtain a common session key, which can be used to ensure the privacy of their later communication.

B. SECURITY ANALYSIS

The security of the key agreement is based on the bilinear Diffie-Hellman (BDH) assumption. According to the proof in [35], the presented protocol can resist both passive attack and active attack.

Theorem 1: According to the proposed protocol, a common session key is derived to ensure the secure communication of nodes.

Proof: The proof of Theorem 1 is with respect to the definition of the block design. In a (7, 3, 1)-design, we have the following characteristics: Each element appears in exactly 3 of the 7 blocks and each block contains 3 elements. In addition, every two elements appear simultaneously in exactly 1 of the 7 blocks. In Step 1, node i in block B_i collects the partial private key from the other 2 nodes in block B_i, which contain 2 partial private keys. In Step 2, node i collects the partial private key from node j if node i is contained in the blocks B_j, which contain 4 partial private keys. Finally, node i collects 6 partial private keys except his own. Moreover, due to the property that every two elements appear simultaneously in exactly once of the 7 blocks, the 6 partial private keys are not repeated. Therefore, a common session key is derived, which is contributed by partial private keys of all nodes. In addition, based on the tree structure, the (7, 3, 1)-design can be further extended to support secure communication among trustworthy nodes in the hierarchical system. □

Theorem 2: The common session key is secure against the passive adversary, which makes a secure communication system for PSN.

Proof: According to the proof in [36], given a random number $y_i \in Z_q^*$, it is hard to distinguish between $(\mathcal{G}, P_{pub}, H_1(ID_i), \hat{e}(\mathcal{G}, e_i r_i \mathcal{S}_i))$ and $(\mathcal{G}, P_{pub}, H_1(ID_i), y_i)$. Similarly, for every node i in PSN, two variables $\hat{e}(\mathcal{G}, e_i r_i \mathcal{S}_i)$ and $\hat{e}(\mathcal{G}, y_i)$ are the same from the viewpoint of the

attacker. Since the common session key is calculated as $\mathcal{K} = \hat{e}(\mathcal{G}, \sum_{i=1}^{7} e_i r_i \mathcal{S}_i)$, given the public parameters $\{p, q, G_1, G_2, \mathcal{G}, \hat{e}, P_{pub}, H_1, H_2\}$, it is hard to distinguish $\mathcal{K} = \hat{e}(\mathcal{G}, \sum_{i=1}^{7} e_i r_i \mathcal{S}_i)$ from the random point $\hat{e}(\mathcal{G}, \sum_{i=1}^{7} y_i)$ in G_2. Therefore, the adversary cannot learn any information about the common session key. □

In addition, our scheme also provides the following properties, which are essential for withstanding active attack.

1) KNOWN SESSION KEY

A known session key prevent the session key held by a fresh participant [37] from being compromised by an adversary, even if the adversary has learned some previous session key. In the presented scheme, the session key r_i is randomly selected by the node in each session. Therefore, an adversary cannot learn any information about the session key of a fresh participant.

2) PERFECT FORWARD SECURITY (PFS) [38]

In our key agreement, the security of the previous session key r_i is based on the elliptic curve discrete logarithm problem(ECDLP) and the BDH assumption. Therefore, the presented scheme provides perfect forward secrecy.

3) KEY CONFIRMATION

If a participant is assured that its counterparts actually have possession of a particular secret key, the protocol provides key confirmation [39]. In the presented key agreement scheme, each node can ensure that its counterparts actually have possession of a common session key.

4) AUTHENTICATION SERVICES

In [40], the homomorphic authenticator is constructed by the Diffie-Hellman shared key. Following the thought, authentication services can be supported in our scheme based on the shared session key among nodes.

VII. CONCLUSION

In a social network that contains a large number of customers, it is difficult to ensure that each customer has an understanding of the rest of the customers because of regional restrictions or different interests. Naturally, a customer will not be able to make a reasonable assessment of unfamiliar customers. In addition, the evaluation will consume a lot of resources in accordance with the general plan. Therefore, based on the (7, 3, 1)-design and tree structure, we propose a hierarchical evaluation system. An accurate trust level of the node in PSN is reflected with respect to the proposed system.

It is worth noting that the proposed hierarchical system can also be used for key agreement in addition to credit evaluation. As a consequence, the session key obtained through the key agreement can be used to ensure secure communication between the trusted nodes. In this paper, an efficient and practical evaluation system is proposed. We believe that the proposed hierarchical evaluation system can establish a perfectly trusted PSN and support secure communication among the trusted nodes in the PSN.

REFERENCES

[1] Z. Yan, M. Wang, V. Niemi, and R. Kantola, "Secure pervasive social networking based on multi-dimensional trust levels," in *Proc. IEEE CNS*, Oct. 2013, pp. 100–108.
[2] L. Li, W. Zeng, Z. Hong, and L. Zhou, "Stochastic Petri net-based performance evaluation of hybrid traffic for social networks system," *Neurocomputing*, vol. 204, pp. 3–7, Sep. 2016.
[3] S.-Y. Chen, C.-F. Lai, R.-H. Hwang, Y.-H. Lai, and M.-S. Wang, "An adaptive sensor data segments selection method for wearable health care services," *J. Med. Syst.*, vol. 39, no. 12, pp. 1–11, 2015.
[4] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-health clouds," *J. Supercomputing*, vol. 72, no. 10, pp. 3826–3849, Oct. 2016.
[5] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.
[6] Z. Yan, Y. Chen, and Y. Shen, "PerContRep: A practical reputation system for pervasive content services," *J. Supercomput.*, vol. 70, no. 3, pp. 1051–1074, Dec. 2014.
[7] E. Sarigöl, O. Riva, P. Stuedi, and G. Alonso, "Enabling social networking in ad hoc networks of mobile phones," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1634–1637, Aug. 2009.
[8] J. Shen, C. Wang, A. Wang, L. Li, Y. Yang, and J. Wang, "Performance comparison of typical routing protocols in ad-hoc networks," in *Proc. Int. Comput. Symp. (ICS)*, 2015, pp. 463–473.
[9] M. W. Macy and J. Skvoretz, "The evolution of trust and cooperation between strangers: A computational model," *Amer. Sociol. Rev.*., vol. 63, no. 5, pp. 638–660, Oct. 1998.
[10] Z. Yan, Y. Chen, and Y. Shen, "A practical reputation system for pervasive social chatting," *J. Comput. Syst. Sci.*, vol. 79, no. 5, pp. 556–572, Aug. 2013.
[11] Z. Yan and M. Wang, "Protect pervasive social networking based on two-dimensional trust levels," *IEEE Syst. J.*, to be published.
[12] K. P. N. Puttaswamy and B. Y. Zhao, "Preserving privacy in location-based mobile social applications," in *Proc. 11th Workshop Mobile Comput. Syst. Appl.*, Feb. 2010, pp. 1–6.
[13] G. Chen and F. Rahman, "Analyzing privacy designs of mobile social networking applications," in *Proc. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Dec. 2008, pp. 83–88.
[14] J. Shen, H. Tan, J. Wang, J. Wang, and S. Lee, "A novel routing protocol providing good transmission reliability in underwater sensor networks," *J. Internet Technol.*, vol. 16, no. 1, pp. 171–178, 2015.
[15] A. Beach, M. Gartrell, and R. Han, "Solutions to security and privacy issues in mobile social networking," in *Proc. Int. Conf. Comput. Sci. Eng.*, Aug. 2009, pp. 1036–1042.
[16] N. Sadeh *et al.*, "Understanding and capturing people's privacy policies in a mobile social networking application," *Pers. Ubiquitous Comput.*, vol. 13, no. 6, pp. 401–412, Aug. 2009.
[17] E. Miluzzo *et al.*, "Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2008, pp. 337–350.
[18] J. Shen, H. Tan, S. Moh, I. Chung, and J. Wang, "An efficient RFID authentication protocol providing strong privacy and security," *J. Internet Technol.*, vol. 17, no. 3, p. 2, 2016.
[19] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2643–2655, Oct. 2015.
[20] X. Li *et al.*, "A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security," *Wireless Pers. Commun.*, vol. 89, no. 2, pp. 569–597, Jul. 2016.
[21] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 37–48, Dec. 2016.

[22] X. Li, K. Wang, J. Shen, S. Kumari, F. Wu, and Y. Hu, "An enhanced biometrics-based user authentication scheme for multi-server environments in critical systems," *J. Ambient Intell. Humanized Comput.*, vol. 7, no. 3, pp. 427–443, Jun. 2016.
[23] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dyn.*, vol. 83, no. 4, pp. 2085–2101, Mar. 2016.
[24] D. He, N. Kumar, H. Shen, and J. Lee, "One-to-many authentication for access control in mobile pay-TV systems," *Sci. China Inf. Sci.*, vol. 59, no. 5, pp. 1–14, 2016, doi: 10.1007/s11432-015-5469-5.
[25] W. Yuan, D. Guan, and S. Lee, "Trust management for ubiquitous healthcare," in *Proc. Int. Symp. Parallel Distrib. Process. Appl. (ISPA)*, Dec. 2008, pp. 63–70.
[26] Y. Karabulut, J. Mitchell, P. Herrmann, and C. D. Jensen, *Trust Management II*. New York, NY, USA: Springer, 2008.
[27] D. N. Kalofonos, Z. Antoniou, F. D. Reynolds, and M. Van-Kleek, "MyNet: A platform for secure P2P personal and social networking services," in *Proc. 6th Annu. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2008, pp. 135–146.
[28] S. M. Allen *et al.*, "Social networking for pervasive adaptation," in *Proc. 2nd IEEE Int. Conf. Self-Adapt. Self-Org. Syst. Workshops (SASOW)*, Oct. 2008, pp. 49–54.
[29] J. Shen, W. Zheng, J. Wang, Y. Zheng, X. Sun, and S. Lee, "An efficient verifiably encrypted signature from weil pairing an efficient verifiably encrypted signature from weil pairing," *J. Internet Technol.*, vol. 14, no. 6, pp. 947–952, 2013.
[30] Y. Lu and J. Li, "A provably secure certificate-based encryption scheme against malicious CA attacks in the standard model," *Inf. Sci.*, vol. 372, pp. 745–757, Dec. 2016.
[31] J. Shen, D.-Z. Liu, C.-F. Lai, Y.-J. Ren, and X.-M. Sun, "A secure identity-based dynamic group data sharing scheme for cloud computing," *J. Internet Technol.*, vol. 99, no. 99, pp. 1–9, 2015.
[32] X. Chen, F. Zhang, W. Susilo, H. Tian, J. Li, and K. Kim, "Identity-based chameleon hashing and signatures without key exposure," *Inf. Sci.*, vol. 265, no. 5, pp. 198–210, May 2014.
[33] O. Lee, S. Yoo, B. Park, and I. Chung, "The design and analysis of an efficient load balancing algorithm employing the symmetric balanced incomplete block design," *Inf. Sci.*, vol. 176, no. 15, pp. 2148–2160, Aug. 2006.
[34] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.
[35] J. Shen, S. Moh, and I. Chung, "Identity-based key agreement protocol employing a symmetric balanced incomplete block design," *J. Commun. Netw.*, vol. 14, no. 6, pp. 682–691, Dec. 2012.
[36] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 213–229, 2003.
[37] S. Blake-Wilson, D. Johnson, and A. Menezes, *Key Agreement Protocols and Their Security Analysis*. Berlin, Germany: Springer, 2006.
[38] J. Li, H. Teng, X. Huang, Y. Zhang, and J. Zhou, *A Forward-Secure Certificate-Based Signature Scheme*. Berlin, Germany: Springer, 2015.
[39] G. Ateniese, M. Steiner, and G. Tsudik, "New multiparty authentication services and key agreement protocols," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 628–639, Apr. 2000.
[40] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *J. Internet Technol.*, vol. 16, no. 2, pp. 317–323, 2015.

JIAN SHEN (M'11) received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2007, and the M.E. and Ph.D. degrees in computer science from Chosun University, Gwangju, South Korea, in 2009 and 2012, respectively.

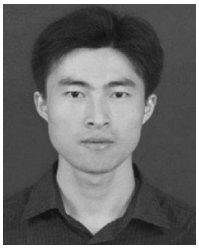
Since 2012, he has been a Full Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include computer networking, security systems, public encryption, and network security.

TIANQI ZHOU received the B.E. degree from Nanjing University of Information Science and Technology, Nanjing, China, in 2016, where she is currently pursuing the master's degree with the School of Computer and Software. Her research interests include computer and network security, security systems, and cryptography.

CHIN-FENG LAI (SM'14) received the Ph.D. degree in the Department of Engineering Science from National Cheng Kung University, Taiwan, in 2008.

He is currently a Professor with the Department of Engineering Science, National Cheng Kung University. His research interests include multimedia communications, sensor-based healthcare, and embedded systems. After receiving the Ph.D. degree, he has authored/co-authored more than 100 refereed papers in journals, conferences, and workshop proceedings about his research areas within four years. He is currently making efforts to publish his latest research in the IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEM ON VIDEO TECHNOLOGY. He is also a Senior Member of the IEEE Circuits and Systems Society and the IEEE Communication Society.

JIGUO LI received the Ph.D. degree from Harbin Institute of Technology in 2003. Since 2003, he has been with Hohai University. He is currently a Professor with the College of Computer and Information Engineering. His major research interests include information security and cryptography, network security, wireless security, and trusted computing. He has published more than 70 scientific papers and two books. He has served as a PC Member of several international conferences and as a Reviewer of international journals and conferences.



XIONG LI received the master's degree in mathematics and cryptography from Shaanxi Normal University in 2009, and the Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications in 2012. He is currently a Lecturer with Hunan University of Science and Technology, China. He has published more than 20 refereed journal papers. His research interests include cryptography and information security.