Peer-to-Peer Netw. Appl. (2016) 9:762–773
DOI 10.1007/s12083-015-0329-z

Practical blacklist-based anonymous authentication scheme for mobile crowd sensing

Hongwei Li · Kun Jia · Haomiao Yang · Dongxiao Liu · Liang Zhou

Received: 31 October 2014 / Accepted: 21 January 2015 / Published online: 8 February 2015

Abstract Mobile crowd sensing (MCS) represents one of the most promising approaches for improving life quality of individuals with sensing and computing devices. MCS is playing a more and more important role in various fields of service, such as traffic monitoring and commercial advertisement. Security and privacy of communication in MCS attract increasing attention from the academia and industry since the sensing data are usually sensitive for users. Some users worry about the leakage of their private information when they share their data to the third parties. To address this issue, in this paper, we propose a practical blacklist-based anonymous authentication scheme in which users can enjoy an anonymous environment and share their information without worrying about any information leakage. Security analysis shows that our scheme can achieve anonymity, blacklistability, nonrepudiation and unlinkability. Performance evaluation demonstrates that our scheme is more efficient in terms of computation overhead compared with the existing works.

Keywords Mobile crowd sensing · Anonymous Authentication · Blacklist-based · Privacy-preserving

H. Li (✉) · K. Jia · H. Yang · D. Liu
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
e-mail: hongweili@uestc.edu.cn

K. Jia
e-mail: jiakunonly@gmail.com

H. Yang
e-mail: haomyang@uestc.edu.cn

D. Liu
e-mail: haohhaha@gmail.com

L. Zhou
National Key Laboratory of Science and Technology on Communication, University of Electronic Science and Technology of China, Chengdu, China
e-mail: lzhou@uestc.edu.cn

H. Li · K. Jia · H. Yang · D. Liu
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China

1 Introduction

The increasing availability of sensors on today's smart phones has already opened up new possibilities for gathering sensed information from our environment. Mobile crowd sensing (MCS) [1, 2] refers to the wide variety of sensing models in which individuals with sensing and computing devices are able to collect and contribute valuable data for third parties. MCS can be deployed on many field applications ,such as cloud computing applications [14, 18, 21], smart grids applications [6, 10–13] and so on. More details, MCS applications can be used to enable a broad spectrum of applications, ranging from monitoring the air pollution condition [3] or location based services to monitoring traffic conditions [4, 5] or social network applications [7, 15]. Figure 1 shows the typical fundamental structure of MCS. The participants use a sensor on mobile to obtain the data which is required about a subject of interest, and upload these data to a tasking entity such as a cloud service, where these data are aggregated, processed and remain available for third parties(e.g., traffic monitoring application or environmental monitoring department) to query and select data of interest. For example, in vehicular networks [9, 16], the traffic monitoring applications, sensors carried by drivers

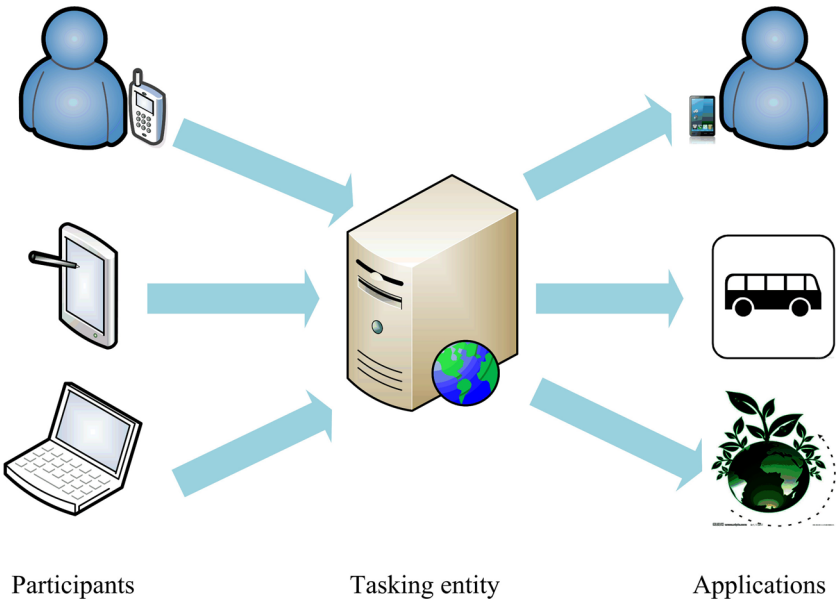


Fig. 1 A typical structure of crowd-sensing application

can collect the information of the traffic condition in real time and real location, such as the traffic flow, traffic jam information and so on, and share these information to the tasking entity(e.g., cloud service), which may be used by third parties (e.g., other drivers) who can select a best transportation route to avoid the traffic jam. Another example of its application is that commercial organizations may be very interested in collecting mobile sensing data to learn more about customer behavior, which demonstrates how useful and beneficial MCS is to our lives. Without doubt, MCS makes our life more convenient and wonderful.

Despite the past nontrivial effort, MCS is still in its infancy, attracting increasing research attention, especially for the privacy protection of users, which is challenging in MCS due to its unique characteristic. For instance, in the traffic monitoring application, as the participants issue the traffic condition and bring much benefit for the third parties, their location and time information will be exposed to the third parties simultaneously. Thus, it is important to design a method to ensure the participants can anonymously access and share the information, which can protect their privacy effectively. Namely, the tasking entity and the third parties users only need to know the traffic conditions, but not know other private information such as participants' IP addresses and identities, etc. On the other hand, participants are not always trustworthy since they may submit false data unscrupulously to earn benefits for themselves. Thus, we need to prevent the misbehavior of participants with respect to the upload of data to tasking entity. Therefore, the proposed scheme can not only protect participants' privacy, but also prevent malicious users from uploading data.

By carefully exploring the intrinsic characteristics of MCS and considering the Quality-of-Experience (QoE) [8] and examining the existing anonymous authentication schemes, we present a Practical Blacklist-based Anonymous Authentication scheme for Mobile Crowd-sensing. Specially, the main contributions of this paper can be summarized as follows:

- ***Practical reputation scores.*** We measure the participant's access authority by his behavior scores which is scored by application provider. Application provider can score each user with a positive or negative score with a category identifier.
- ***Blacklisting malicious users.*** We propose a technique that can prevent the malicious users' misbehavior. In our scheme, the application provider provides a series of policies that participant's reputation scores must satisfy. Otherwise, the participant will be added in the blacklist, therefore the participant cannot enjoy the anonymous environment of MCS. In particular, our scheme can add misbehaved users to a blacklist using their pseudonyms instead of their real identity, thereby preventing the potential privacy leakage.
- ***Efficiency.*** Our scheme takes full consideration of the computational ability of mobile devices. The proposed scheme is mainly based on the symmetric-key encryption to achieve anonymous authentication instead of the cost-heavy public-key encryption or pairing. Thus, the proposed scheme is efficient and particularly suitable for resource-constrained mobile clients of MCS.

The remainder of this paper is organized as follows. In Section 2, the system model and security requirements

are formalized. We present notations and cryptographic primitives in Section 3. In Section 4, we propose our scheme. Followed by the security analysis and performance evaluation of our scheme in Section 5 and Section 6, respectively. We present the related work in Section 7. Finally, we conclude this paper in Section 8.

2 System model and security requirements

2.1 System model

As shown in Fig. 2, there are four types of entities involved in our scheme: participant, pseudonym manager(PM for short), application provider(AP for short) and network manager(NM for short).

- *Participant*: Participant is the entity that measures and shares required data about a subject of interest to the application provider by using sensors on everyday devices, such as smart phones, personal digital assistant(PDA). However, not all participants are always honest since some participants maybe misbehave.
- *Pseudonym Manager(PM)*: PM is in charge of mapping the participant's resources id(e.g., IP address, MAC address), to the pseudonyms. PM is the first entity that the participant must contact, which determines whether the participant is permitted to register or not. As a result, PM's duties are limited to determining the right of registration and mapping IP address to pseudonym.

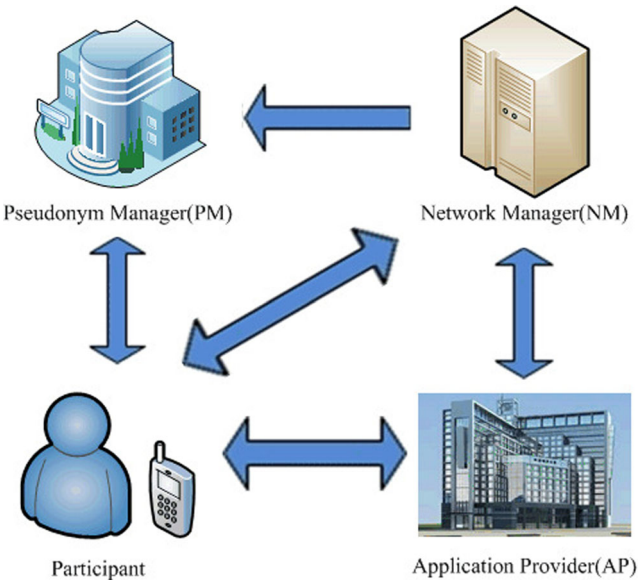


Fig. 2 System model for anonymous authentication scheme

- *Application Provider(AP)*: AP not only provides the services to the user but also manages the reputation scores of a participant, including scoring grades, updating scores and modifying scores. In addtion, AP maintains a blacklist which is used to determine participant's right of enjoying the anonymous environment of MCS. AP lays down the policies that each participant must satisfy, otherwise, the participant cannot enjoy the anonymous environment.
- *Network Manager(NM)*: NM is the control center of the system. NM initializes the system parameters, such as secret keys and secure hash functions, and issues them to other entities. Moreover, NM is in charge of maintaining and computing the participant's current scores and generating the participant's credentials in order to authenticate with AP. As we will explain, NM only knows which AP that participant wants to contact, the other information, such as the participant's IP address, is kept unknown.

2.2 Security requirements

We present informal definitions of the desired security properties. The security requirements in our scheme should cover these four aspects.

- *Anonymity*: Adversary can get only a train of pseudonyms of participants instead of the real identity based on the existing communication.
- *Blacklistability*: Only a participant who is not in the black list can obtain the anonymous service. In other words, a participant cannot obtain the service if he has been put in the backlist.
- *Nonrepudiation*: The property of nonreputation is a fundamental requirement for our scheme. Namely, a participant cannot deny that he has accessed the service provided by AP.
- *Unlinkability*: All of the messages generated by a participant should not leak any information to an adversary by allowing the adversary to trace them.

3 Preliminaries

3.1 Notation

For easier illustration, Table 1 lists some important notations which will be given further explanation where they occur for the first time.

4 Proposed scheme

In this section, we propose our scheme, which mainly consists of the following five phases: System Initialization,

Table 1 Notation

$\mathscr{F}$	function of generating $Seed_t$
$\mathscr{G}$ / g	function of generating $Psdm_t$ / $Psdm_0$
Encrypt	symmetric encryption function
Decrypt	symmetric decryption function
$\mathscr{H}_0$ / $\mathscr{H}_1$	secure hash function $\mathscr{H}_0$ / $\mathscr{H}_1$
Enc / Dec	public encryption / decryption function

Participant Registration, Authentication, Update scores and Modify scores.

4.1 System initialization

During system setup, NM interacts with other entities. First, NM generates a number of private keys for the system. The private key $macKey_{NP} \in \{0,1\}^m$, which is used to generate the pseudonym for PM and verify the integrity of pseudonym for NM, and then NM issues it to PM through a secure communication channel. The key $Key_{NA} \in \{0,1\}^m$ shared with AP is used by NM and AP to exchange information secretly, and send them to AP over a secure channel. In addition, NM generates the private key $seedKey_N \in \{0,1\}^m$,which is used to generate the seed, the private key $encKey \in \{0,1\}^m$ is used to generate the $SCUP$. At the same time, NM applies a public key pair $(PriKey_N, PubKey_N)$, where $PriKey_N \in \{0,1\}^m$ is the private key, and the $PubKey_N$ is the corresponding public key. NM also picks a number of secure hash functions. The secure hash function $\mathscr{H}_0 : \{0,1\}^* \times \{0,1\}^m \rightarrow \{0,1\}^m$ to generate the pseudonym for PM, and the secure hash function $\mathscr{H}_1 : \{0,1\}^m \times \{0,1\}^* \times \mathbb{Z}_q{}^* \times \{0,1\}^m \rightarrow \{0,1\}^m$ to generate the $seed_0$. Finally, NM publishes these hash functions $\mathscr{H}_0$ and $\mathscr{H}_1$.

4.2 Participant registration

In our scheme, a participant must use a ticket which is requested from NM to authenticate. As illustrated in Fig. 3, the transaction is divided into linkability windows of duration w, each of which is split into m transaction identifiers. Particularly, a participant must register once in each linkability window, namely, after accessing to AP m times, the participant must register again.

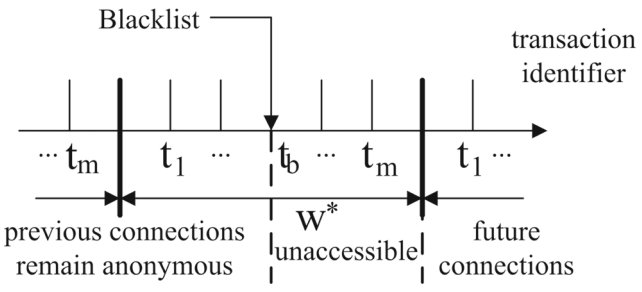


Fig. 3 The life cycle of anonymous authentication

4.2.1 Create the pseudonym

A participant with identity uid must apply for a pseudonym $ctpd$ from PM firstly. A pseudonym $ctpd$ consists of nym and mac: nym is a hash mapping of the participant's identity (e.g., IP address or other resources), the linkability window W_{crt} for which the pseudonym is valid, mac is used to verify the integrity of pseudonym by NM. We suppose PM owns a long-term secret $pmKey_p \in \{0,1\}^m$, which is used to generate the nym. Therefore, after receiving uid from the participant, PM calculates the participant's $ctpd$ as below algorithm.

$$\begin{aligned} nym &= \mathscr{H}_0(uid||W_{crt}, pmKey_p) \\ mac &= \mathscr{H}_0(nym||W_{crt}, macKey_{NP}) \\ ctpd &= (nym, mac) \end{aligned} \tag{1}$$

where the $macKey_{NP}$ is the secret key shared with PM and NM. After successfully generating $ctpd$, PM sends it to participant.

4.2.2 Verify the pseudonym

Participant must register to NM to get the ticket to authenticate. Thus, the participant sends the $ctpd$ and sid to the NM for registering, where sid is the identity of AP which participant wants to access. After receiving $ctpd$, NM firstly checks its freshness and integrity. Thus, NM reads the current linkability window as W_{crt}, which guarantees the freshness of pseudonym, and then extracts the nym and mac from the $ctpd$ to check the integrity of pseudonym is as following.

$$mac = \mathscr{H}_0(nym||W_{crt}, macKey_{NP}) \tag{2}$$

NM accepts the pseudonym if and only if the above formula is tenable, otherwise, terminates the scheme with failure.

4.2.3 Create the participant's score list and credential

In order to save a participant's reputation scores, NM maintains a score list $\mathscr{L}$ for the participant. The list $\mathscr{L}$ can correctly record different participants scores.

Table 2 Scores list data structure

$Psdm_0$	S_1	S_2	$\cdots$	S_L	t_1	t_2	$\cdots$	t_k

As shown in Table 2, the score queue $\mathscr{L}$ consists of three parts: the unique identifier, the previous score and the recent K transactions.

- ***The Unique Identifier***: The first value($Psdm_0$) is the unique identifier to distinguish various participants. For the same participant, the $Psdm_0$ is different if the AP which the participant wants to access is different, since $Psdm_0$ is a mapping value of sever identity sid. Using the unique identifier $Psdm_0$, NM can correctly compute and update the participant's scores. The more explanation of $Psdm_0$ as following.
- ***The Previous Score***: The middle L values represent a participant's current scores of L categories. In our scheme, AP scores a participant's behavior from different perspectives. In our scheme, each transaction identifier is associated with L scores.
- ***The Recent K Transactions***: In order to prevent NM from updating the participant score repeatedly, the recent K transaction identifiers are reserved.

The score list L is initialized to null, that is, besides the unique identifier, the previous scores and the recent K transactions are all initialized by null.

Particularly, the participant must provide a valid ticket which is acquired as part of a credential from the NM to AP for authentication at each time. The pseudonym $Psdm$ in the tickets which serves as an identifier for a particular time period, and the $Psdm$ is evaluated from the seeds.

Seed as a parameter evolves throughout a linkability window using a seed-evolution function $\mathscr{F}$, the seed for the next transaction identifier $seed_{next}$ is computed from the seed for the current linkable window $seed_{cur}$, that is:

$$seed_{next} = \mathscr{F}(seed_{cur}) \tag{3}$$

The first seed ($seed_0$) is generated by hashing a participant's pseudonym $ctpd$, the identity sid of the server, the linkability window W_{crt} for which the seed is valid, and the secret key $seedKey_N$ of NM. As a consequence, a seed is useful just for a particular AP to access a particular participant during a particular linkability window.

$$seed_0 = \mathscr{H}_1(ctpd, sid, W_{crt}, seedKey_N) \tag{4}$$

$Psdm$ as an identifier is used to authenticate with AP. Just like the generation of a seed, the Psdm($Psdm_t$) for a certain linkable window is generated from the corresponding seed($seed_t$) by applying the psdm-evaluation function $\mathscr{G}$ as following:

$$Psdm_t = \mathscr{G}(seed_t) \tag{5}$$

However, NM calculates the first corresponding $Psdm_0$ by using the psdm-evolution function g as: $Psdm_0 = g(seed_0)$. Thus, every $Psdm$ is only associated with one $seed$. Obviously, without a seed, adversary can not generate the the sequence of $Psdm$.

A credential must be provided by a participant. Particularly, a ticket is only used once for a particular linkability window. NM reads the current linkable window as w, and the computation of tickets and a credential for a participant is as algorithm-1: where the $SCUP$ is the encrypted data. Especially, $SCUP$ contains the first $Psdm(Psdm_0)$, with which NM can easily update the scores or add a misbehaved participant to the blacklist. Particularly, the $score = S_1 \| S_2 \| \cdots \| S_L$, where S_i represents the ith score. In addition, each ticket contains two parts: CT and TV. Particularly, TV is the encrypted data of CT, which not only protects the privacy of information, but can also verify the integrity of the $Psdm$ ticket for AP. After successfully computing the $Cred$, NM signs it by executing the encrypt function Enc to ensure the security of the data.

$$Cred_{Sig} = Enc(Cred, PriKey_N) \tag{6}$$

Algorithm 1 generate tickets and credential

Input: The first seed $seed_0$, linkable window w, AP's identity sid and participant's scores $score$

Output: $Cred$

```
for t = 1 to m do
    seed_t = F(seed_{t-1})
    Psdm_t = G(seed_t)
    SCUP_t = Encrypt(Psdm_0||t_i, encKey)
    CT = sid||w||Psdm_t||SCUP_t||t_i||score
    TV = Encrypt(CT, Key_NA)
    ticket = CT||TV
    Tickets[t] = ticket
end for
Cred = Psdm_0||Tickets return Cred
```

where $PriKey_N$ is NM's secret private key especially. Finally, NM issues $< Cred, Cred_{Sig} >$ to the participant, where the $Cred_{Sig}$ is the signature of $Cred$. Figure 4 describes the message flow of the creating and verifying the participant's credential phase.

From above, we can easily see that as long as the PM and NM do not collude, the system cannot identify which participant is connecting to which server; the NM only knows the pseudonym-server pair and the PM only knows the user identity-pseudonym pairs.

4.3 Authentication

A participant must provide a valid ticket, which is acquired as part of a credential from the NM to authenticate. Firstly, the participant must check whether it is in the blacklist or

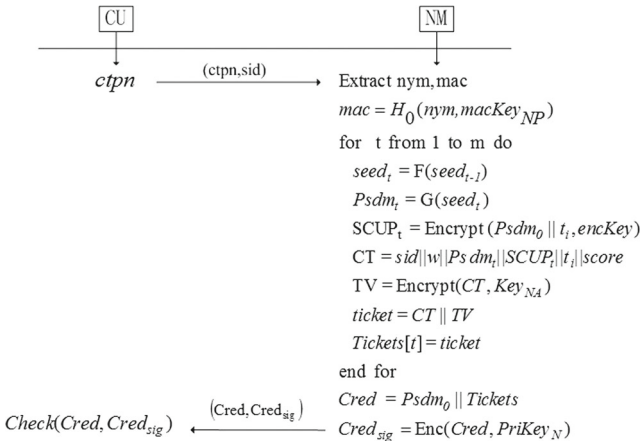


Fig. 4 The process of creating and verifying credential

not. If a participant finds that it is in the blacklist unfortunately, as an honest participant, it terminates the scheme immediately. Otherwise, the participant sends its tickets to AP, and AP passes the participant's authentication if the participant's reputation scores satisfy the policy. We should note that the whole procedure take place under the anonymous condition, namely AP knows nothing but a series of scores.

4.3.1 Check whether blacklisted

Participant must verify the signs firstly to guarantee the integrity of $Cred_{Sig}$. The process of checking is as following:

$$Cred = Dec(Cred_Sig, PubKey_N) \tag{7}$$

where the $PubKey_N$ is the public key of NM. If the above equation is established, participant then checks whether it is in the blacklist via comparing with the $Psdm$ value in the blacklist or not. The blacklist is composed of a series of $Psdm_0$, the current linkability window W_{crt} and the current transaction identifier. Especially, the current linkability window and the current transaction identifier guarantee the freshness of the blacklist. If the participant's $Psdm_0$ is in the blacklist, then the authentication procedure is terminated and is regarded as a failure. Otherwise, the participant must extract the $tickets$ from the $Cred$ and sends it to AP.

4.3.2 Check whether the policy is satisfied

When the AP receives the $tickets$, it checks the validity of tickets as following.

– ***The validity of tickets***: AP extracts the TV and CT from the tickets, performs the following operations to check the integrity of tickets:

$$CT = Decrypt(TV, Key_{NA}) \tag{8}$$

– ***The freshness of tickets***: AP reads the current linkability as W_{now}, and compares it with the w in CT. If W_{now} equals to w, it proves that the tickets are fresh.

AP rejects the request if the ticket is not valid or fresh. Otherwise, AP checks the participant's score to see whether it satisfies the policy, which is a boolean combination of scores category-threshold and is stored in AP database. Particularly, AP formulates a policy ply which is made of several different sub-policies ply_i, and a sub-policy consists of boolean combinations of L scores, i.e.,

$$\begin{aligned} ply &= ply_1 \vee ply_2 \vee ply_3 \vee \cdots \vee ply_n \\ ply_i &= S_{p_1} \| S_{p_2} \| \cdots \| S_{p_L} \end{aligned} \tag{9}$$

Every participant must satisfy one of sub-policies. Therefore, AP extracts scores from the tickets, and proves that the participant's every category score is less than one of corresponding sub-policy scores. That means, there exists an integer k: $ply_k = S_{p_1} \| S_{p_2} \| \cdots \| S_{p_L}$ and for each value i for i=1 to L satisfies

$$S_i > S_{p_i} \quad 1 \le i \le L \tag{10}$$

Where S_i presents the participant's current ith score. If the participant's scores satisfy the policy, AP provides the service to the participant. From above, we can easily conclude that AP does not know the participant's real identity.

4.3.3 Add participant to the blacklist

Sometimes, the participant's scores do not satisfy any policy unfortunately, thus, AP will add the participant to the blacklist and not provide it anonymous service anymore. However, AP does not know any of the participant's information due to the anonymity property of the authentication procedure. Therefore, AP applies to NM for the participant's first pseudonym($Psdm_0$) to add it to blacklist. AP generates a boolean value $black$ and sets it true, which represents that participant's score doesn't satisfy the policy. AP sends the TB which is the encrypted data of $ticket$ and boolean value $black$ to NM as follows:

$$TB = Encrypt(ticket||black, Key_{NA}) \tag{11}$$

where Key_{NA} is the secret key shared by NM and AP. As we know, only NM can decrypt TB, which guarantees the security of information. When receiving the "black" message, NM computes the $Psdm_0$ for AP. First of all, NM decrypts the data of TB and extracts the $SCUP_t$ to compute the $Psdm_0$ as follows:

$$\begin{aligned} ticket||black &= Decrypt(TB, Key_{NA}) \\ Psdm_0 &= Decrypt(SCUP_t, encKey) \end{aligned} \tag{12}$$

After computing $Psdm_0$ successfully, NM sends it to AP. For security, NM firstly encrypts it as follows:

$$PB = Encrypt(Psdm_0||tickets, Key_{NA}) \tag{13}$$

When receiving the encrypted $Psdm_0$, AP decrypts it as $Psdm_0||ticket = Decrypt(PB, Key_{NA})$. And NM adds $Psdm_0$ to his black list. Thus, the malicious participant is added into the blacklist successfully. Figure 5 describes the message flow of the adding malicious users to the blacklist.

4.4 Update scores

AP can score any behavior of a participant. For any special transaction identifier, AP can give it a proper reasonable score. Let us say the score is: $UPSCORE = S_{UP_1}\|S_{UP_2}\|\cdots\|S_{UP_L}$ where the S_{UP_i} represents the ith category score. AP sends the scores $UPSCORE$, transaction identifier t_i and $Psdm_t$ to NM for updating. AP extracts the $Psdm_t$ and the transaction identifier t_i, and encrypts it as follows:

$$\begin{aligned} &US = Psdm_t\|t_i\|UPSCORE \\ &UPSC = Encrypt(US, Key_{NA}) \end{aligned} \tag{14}$$

where the $UPSCORE$ are scores of participants scored by AP. Only NM can decrypt $UPSC$ since only NM knows the secret key Key_{NA}.

NM updates the scores for the participant. When receiving the $UPSC$, NM checks its validation as shown bellow:

$$US = Decrypt(UPSC, Key_{NA}) \tag{15}$$

Assume the above equation is tenable, then, NM extracts the t_i from the US, and checks the value via comparing with the recent K transaction identifier. If the t_i is in the recent K transactions, NM extracts the score as bellow.

$$S_{UP_1}\|S_{UP_2}\|\cdots\|S_{UP_L} = UPSCORE \tag{16}$$

for $i = 1$ to L,

$$S_i = S_i + S_{UP_i} \tag{17}$$

After this computation, NM successfully updates the participant's score in the score list $\mathscr{L}$.

4.5 Modify scores

If at some later time, AP desires to upgrade a score for transaction identifier t. Let the original score be S_1, S_2, $\cdots$, S_L, and the updated score be S'_1, S'_2, $\cdots$, S'_L, and the difference is d, that means the value $d_i = S'_i - S_i$. Therefore, AP just only sends the value d_i to NM to upgrade the score. Let the L categories update score $d = d_1\|d_2\|\cdots\|d_L$. AP creates a boolean value MD, and sets it to true. AP encrypts the $Psdm_t$ and the value d and sends them to NM. Let $MDSC = Psdm_t\|d\|MD$, AP uses the shared key to encrypt it and sends to NM. That is

$$EMSC = Encrypt(MDSC, Key_{NA}) \tag{18}$$

Upon receiving the $EMSC$, NM firstly checks if the message is valid by following operations.

$$MDSC = Decrypt(EMSC, Key_{NA}) \tag{19}$$

NM extracts the boolean value MD and the difference of scores d. The boolean value MD represents that AP request to modify the previous scores. Therefore, NM adds the different score d_i to the previous score S_i as follows:

$$\begin{aligned} &For\ i = 1\ to\ L \\ &S_i = S_i + d_i \end{aligned} \tag{20}$$

Thus, NM successfully upgrades the scores.

5 Security analysis

This section presents the security analysis of our authentication scheme. Our analysis will focus on how our scheme achieves anonymity, blacklistability, nonrepudiation and unlinkability.

5.1 Anonymity

Anonymity means that an adversary cannot obtain the real identity of any participant based on the existing communication. In the tickets presented by a participant, only $Psdm_t$ and $SCUP_t$ are functions of the user's identity. However,

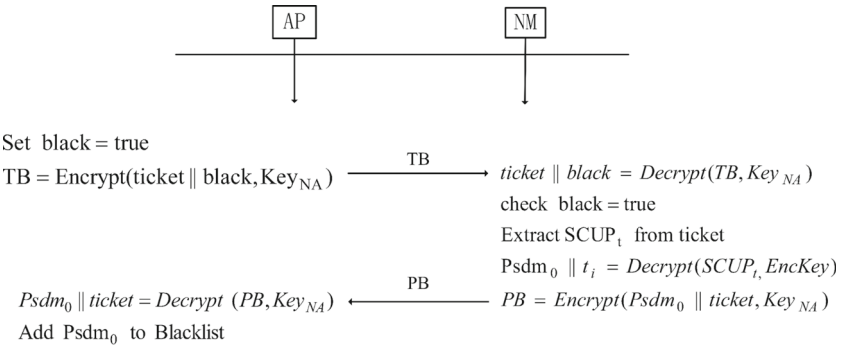


Fig. 5 The process of adding malicious users to blacklist

since the adversary has not obtained any seed for the user, $Psdm_t$ is a series of pseudonyms, so adversary can get nothing from $Psdm_t$. Moreover, because adversary does not know the NM secret key of encryption $SCUP_t$ and the security of AES, the adversary still cannot get any information of a participant. Furthermore, assume an adversary gets $Psdm_0$ that is issued to AP when a participant is blacklisted. However, the AP does not know the hash function and parameters of generating $Psdm_0$, so AP cannot get any information of the participant either. Thus, our proposed scheme can fully exert its ability of protecting a participant's anonymity.

5.2 Blacklistability

It is easy to show that if each participant has been blacklisted in any previous transaction of the current linkability window, the participant cannot authenticate in the current transaction. AP adds $Psdm_0$ to the blacklist when a participant misbehaves. Next time, the participant terminates authentication when he finds his $Psdm_0$ is in the blacklist. At this point, as an honest participant, he will terminate authentication in that circumstance. However, in the real world, the participant may be not honest, so he connects to AP continually by using the tickets which are issued by NM. As long as NM and AP are honest, AP will terminate authentication as well. Namely, upon receiving the tickets, AP checks the participant's scores to see whether or not it satisfies the policy, then terminates authentication if it fails. From the above argument we can draw a conclusion that our scheme meets blacklistablility perfectly.

5.3 Nonrepudiation

After a successful access, a participant cannot deny that he has accessed the service provided by AP. In our scheme, the tickets issued by NM blend participant and AP together. AP and NM use a secure hash function to map a unique source identity to a pseudonym. Each ticket evolves from the same $seed_0$ though NM issues many tickets for different transactions. Moreover, Since the security of hash function, every participant is associated with a unique $seed_0$, and every $seed_0$ is associated with a series of specific pseudonyms. As discussed above, we can conclude that there is no chance for a participant to deny that he has accessed the service.

5.4 Unlinkability

Unlinkability means that all sessions generated by a participant should not leak any information to the adversary. We assume the contrary that an adversary gets enough information so that he can distinguish all participants. From the adversary's perspective, we can separate all of the tickets into two groups, one set of all the tickets come from the same participant, and the other one of all tickets come from different participants. In our scheme, every participant owns a unique $seed_0$, then generates a series of different tickets, thus the same participants has tickets that are different in different transactions. For different participants, because of the security of hash function, their tickets are different as well. Apparently, there isn't significant evidence to prove that whether all tickets collected by the adversary come from one participant or not. We can easily conclude that our scheme meets unlinkability firmly.

We carefully select three existing schemes for comparative analysis and the results are summarized in Table 3. We discuss security properties of the above four aspects in these schemes respectively. The scheme in [27] only has the property nonrepudiation and unlinkability. In [19], only nonrepudiation property is satisfied. In LZCK [28], besides the property of blacklistability, the other properties are achieved. From Table 3, we can conclude that our new scheme achieves a higher security level with strict anonymity and other properties.

6 Performance evaluation

In this section, we evaluate the performance of the proposed scheme in terms of functionality and computation. In addition, we implement our scheme and gain the computation time of each phase. Especially, We compare our scheme with the LZCK [28] in the authentication phase.

6.1 Functionality

The basic aim of our proposed scheme is protecting privacy of the participant, as a result, we implemented the anonymous authentication technology. However, that technology brings a series of problems that the malicious participant abuse the environment of MCS. Therefore, we propose a practical and scalable scheme to support blacklist-based

Table 3 Comparison of security properties among different schemes

x	[27]	[19]	LZCK[28]	Our scheme
A			√	√
B			√	√
N	√	√		√
U	√		√	√

Note: A is the property of anonymous, B is the property of blacklistability, N is the property of nonrepudiation, U is the property of unlinkability

anonymous authentication. Especially, our scheme can limit the participant's right to enjoy anonymous service. In our scheme, the participant's right is measured by his own reputation scores which are scored with the participant's previous behavior by AP with a positive or negative score on the score list. This method not only achieves limiting the participant's right in the anonymous environment but also it is most practical and scalable for implementation on MCS.

6.2 Computation overhead

We have identified the major operations for each of the schemes as shown in Table 4. In particular, we list the run time of symmetric key cryptography and security hash function considering the limited computation ability for the participant of CMS(e.g. smartphones). The symbols T_{rs} and T_{rv} represent the time cost of RSA sign and verification, respectively. The symbols T_{se} and T_{hs} represent respectively the time cost of the symmetric-key cryptography and cryptographic hash computation.

Particularly, we carefully select a scheme LZCK [28] and compare the computation of authentication. From the Table 5 we can see that the main computation cost in [28] is exponentiation in $\mathbb{Z}_q{}^*$, multiplication in $\mathbb{G}_1$ and pairing. And the symbols T_{ex}, T_{ml} and T_p represent them respectively.

From Tables 4 and 5, we can easily draw a conclusion of our scheme's computation complexity. In the *participant Registration* phase, total computation overhead of our scheme is $4T_{hs} + 2T_{se} + T_{rs}$ at PM and NM. However, the computation of LZCK [28] is $2T_p + T_{rs}$. In the *Authentication* phase for our scheme at participant, the computation overhead is only T_{rv}, and at AP is only T_{se}. However, In LZCK [28], the computation at the participant is $4T_{ml} + T_{ex} + 2T_{hs}$, and the computation at AP is $T_p + 2T_{ml} + T_{ex} + 2T_{hs}$. In *Authentication* phase, the computation overhead is $3T_{se}$ for both AP and NM respectively if the participant will be added in the blacklist. Otherwise, the computation overhead is only T_{se}. The other phases, including *Update scores* phase, *Modify scores* phase, the computation overhead is T_{se} for both NM and AP. The comparison of computation overhead is shown in Table 5. Consequently, our scheme is more efficient than LZCK [28].

To evaluate the computation overhead of the proposed schemes, we have implemented our scheme in C++. It uses the famous MIRACL library for the cryptographic operations. Especially, we use SHA-256 for the cryptographic hash functions, AES-256 in CBC-mode for the symmetric encryption Enc; 1,024-bit RSA for the digital signatures Sig. The simulation environment of AP is Windows XP OS over an Inter(R) Pentium IV 2.56GHz processor and 2GB memory. The hardware environment of the mobile crowd sensing has a low-power high-performance 32-bit Inter(R) PXA270 624MHz processor and 128MB memory running Windows CE 5.2 OS. For each experiment, we report the average of 10 runs.

The experiment result of our scheme is shown in Fig. 6, we can easily see that the run time grows linearly as the number of entries increases. It takes about 155ms to authenticate when the entity is 1000, which only occurs when the participant's reputation scores don't satisfy the policy. Otherwise, the time of authentication at AP will be less, and the run time is only about 50ms. In *Update scores* phase and *Modify scores* phase, the run time at NM and AP is about 50ms when entity is 1000, since the operations at NM and AP are similarity. Especially, we compare the run time at AP and the participant between LZCK [28] and our scheme for authentication phase. The results as shown in Fig. 7 demonstrate that our scheme generally outperforms LZCK [28], since Liu's scheme is mainly based on bilinear pairing and our scheme is based on symmetric key cryptography. Therefore, these make it more suitable to implement for mobile crowd sensing.

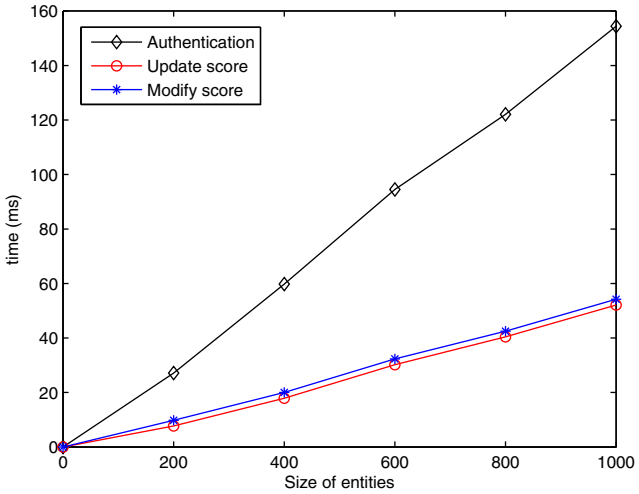
Table 4 Complexity analysis for every phase

Phases	Parties	Computation
Participant Registration	PM	$2T_{hs}$
	NM	$2T_{hs} + 2mT_{se} + T_{rs}$
	Participant	$2T_{rv}$
Authentication	AP	$3T_{se}$
	NM	$3T_{se}$
	AP	T_{se}
Updating scores	NM	T_{se}
	AP	T_{se}
Modify scores	NM	T_{se}

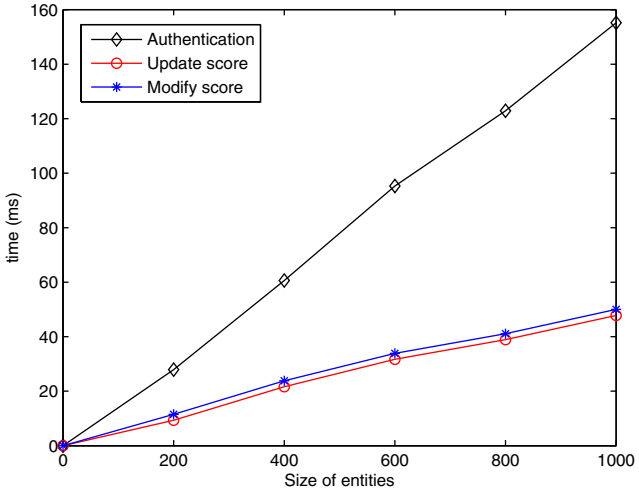
Note: m is the number of transaction for each linkablity window. Particularly, the authentication phase, the value in the table is the situation of a participant who will be added in blacklist. Otherwise, the computation is T_{se} especially

Table 5 Comparison of Computation overhead

		Our scheme	LZCK [28]
Participant Registration		$4T_{hs} + 2T_{se} + T_{rs}$	$2T_p + T_{rs}$
Authen-tication	Participant	T_{rv}	$4T_{ml} + T_{ex} + 2T_{hs}$
	AP	T_{se}	$T_p + 2T_{ml} + T_{ex} + 2T_{hs}$



(a) The run time at NM



(b) The run time at AP

Fig. 6 The performance at a(NM) and b(AP) of various phases

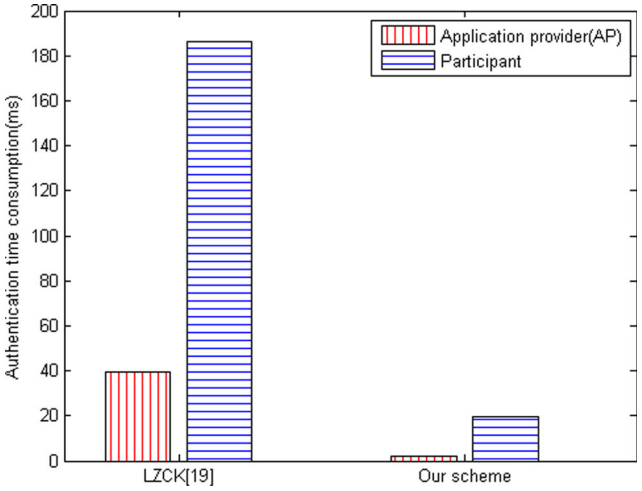


Fig. 7 A comparison of running time between different schemes

7 Related work

Theoretically, anonymous authentication in MCS can be implemented by traditional public-key cryptosystems (PKC) [17, 19, 20, 22]. In particular, Yang et al. [17], presents a novel password-based remote user authentication scheme using bilinear pairings by introducing the concept of private key proxy quantity. However, the computational cost on the user side is a critical issue for implementation on MCS. In [19], propose an ID-based remote mutual authentication with key agreement scheme on ECC, which is based upon the ID-based concept, and the proposed scheme does not require public keys for users so that the additional computations for certificates can be reduced. In addition, which have better performance, thanks to the smaller key size used in ECC. For example, 160-bit ECC achieves the same security level as 1,024-bit RSA. However, as most other PKC schemes, which requires a certification authority (CA) to maintain a pool of certificates for users' public keys, and the users need extra computation to verify the certificates of others. In [20], presents a mutual authentication and key exchange scheme using bilinear pairings, which is based on the computational Diffie-Hellman assumption and the random oracle model, but this design may cost a bit of high computational and not available to implement on MCS. Therefore most of the designs are infeasible in mobile networks, because PKC needs to compute modular exponentiation which may consume more computational resource than what mobile devices can offer.

In order to prevent the abuse of anonymous environment, a few schemes [23–26] have been proposed to revoke access for misbehaving users while maintaining their anonymity. Especially, in [25], users must prove in zero knowledge that each entry on the blacklist does not correspond to an authentication made earlier using their credential, resulting in authentication times linear in the size of the blacklist, which, obviously, cannot suit for mobile device. In [23], Lin et al. propose a scalable anonymous black-listing scheme based on bilinear pairings. Obviously, that may consume much computational resource which is beyond what mobile devices can offer. Therefore, most of these schemes can not be easily implemented on MCS.

8 Conclusions

In this paper, we have utilized the blacklist technique to propose a practical anonymous scheme to preserve privacy of MCS participants when they make access to MCS terminals. Detailed secure analysis shows the proposed scheme can satisfy the desirable security requirements. Performance

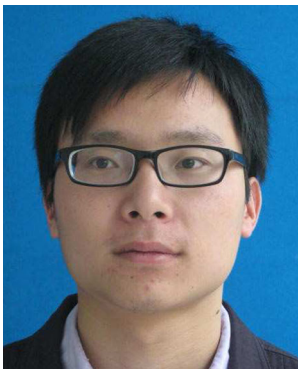
evaluation shows that the proposed scheme can achieve better efficiency in terms of computation overhead compared with the existing works.

Acknowledgments This work is supported by the National Natural Science Foundation of China under Grants 61472065, 61350110238, 61103207, U1233108, U1333127, and 61272525, the International Science and Technology Cooperation and Exchange Program of Sichuan Province, China under Grant 2014HH0029, China Postdoctoral Science Foundation funded project under Grant 2014M552336, and State Key Laboratory of Information Security Open Foundation under Grant 2015-MS-02.

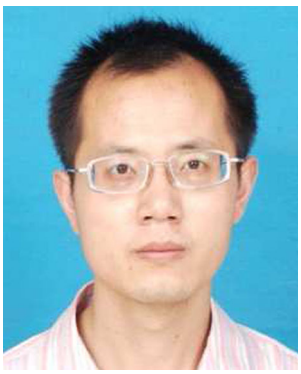
References

1. Ganti RK, Ye F, Lei H (2011) Mobile crowdsensing: current state and future challenges. IEEE Commun Mag 49(11):32–39
2. Cheung M, Hou F, Huang J (2014) Participation and reporting in participatory sensing, the 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp 357–364
3. Honicky R, Brewer EA, Paulos E, White R (2008) N-smarts: networked suite of mobile atmospheric real-time sensors. In: Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions, pp 25–30
4. Mohan P, Padmanabhan VN, Ramjee R (2008) Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: Proceedings of the 6th ACM conference on Embedded network sensor systems, pp 323–336
5. Talasila M, Curtmola R, Borcea C (2013) Improving location reliability in crowd sensed data with minimal efforts. In: The 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC). IEEE, pp 1–8
6. Liu D, Li H, Yang Y, Yang H (2014) Achieving multi-authority access control with efficient attribute revocation in smart grid. In: IEEE International Conference on Communications (ICC). IEEE, pp 634–639
7. Froehlich J, Dillahunt T, Klasnja P, Mankoff J, Consolvo S, Harrison B, Landay JA (2009) Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp 1043–1052
8. Mianxiong D, Kimata T, Sugiura K, Zettsu K (2014) Quality-of-experience (qoe) in emerging mobile social networks. IEICE Trans Inf Syst 97(10):2606–2612
9. Ota K, Dong M, Zhu H, Chang S, Shen X (2011) Traffic information prediction in urban vehicular networks: A correlation based approach. In: IEEE Wireless Communications and Networking Conference (WCNC). IEEE, pp 1021–1025
10. Li H, Yang Y, Wen M, Luo H, Lu R (2014) Emrq: An efficient multi-keyword range query scheme in smart grid auction market. KSII Trans Internet Inf Syst 11:8
11. Yang Y, Li H, Wen M, Luo H, Lu R (2014) Achieving ranked range query in smart grid auction market. In: IEEE International Conference on Communications (ICC). IEEE, pp 951–956
12. Li H, Lin X, Yang H, Liang X, Lu R, Shen X (2014) EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key Evolution in Smart Grid. IEEE Trans Parallel Distrib Syst 25(8):2053–2064
13. Li H, Lu R, Zhou L, Yang B, Shen X (2014) An Efficient Merkle Tree Based Authentication Scheme for Smart Grid. IEEE Syst J 8(2):655–663
14. Li H, Dai Y, Tian L, Yang H (2009) Identity-based authentication for cloud computing. In: Cloud computing. Springer, pp 157–166
15. Reddy S, Parker A, Hyman J, Burke J, Estrin D, Hansen M (2007) Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype. In: Proceedings of the 4th workshop on Embedded networked sensors. ACM, pp 13–17
16. Ota K, Dong M, Chang S, Zhu H (2014) Mmcd: Max-throughput and min-delay cooperative downloading for drive-thru internet systems. In: IEEE International Conference on Communications (ICC). IEEE, pp 83–87
17. Yang C, Ma W, Wang X (2007) Novel remote user authentication scheme using bilinear pairings. In: Autonomic and Trusted Computing. Springer, pp 306–312
18. Yang Y, Li H, Liu W, Yang H, Wen M (2014) Secure Dynamic Searchable Symmetric Encryption with Constant Document Update Cost. In: Proceedings of GLOBECOM. Anaheim, California, USA. to appear
19. Yang J-H, Chang C-C (2009) An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Comput Secur 28(3):138–143
20. Tseng Y-M, Wu T-Y, Wu J-D (2007) A mutual authentication and key exchange scheme from bilinear pairings for low power computing devices. In: The 31st Annual International Computer Software and Applications Conference. IEEE, pp 700–710
21. Li H, Liu D, Dai Y, Luan T, Shen X Enabling Efficient Multi-keyword Ranked Search over Encrypted Cloud Data through Blind Storage. In: IEEE Transactions on Emerging Topics in Computing, doi:10.1109/TETC.2014.2371239
22. Ren J, Harn L (2013) An efficient threshold anonymous authentication scheme for privacy-preserving communications. IEEE Trans Wirel Commun 12(3):1018–1025
23. Lin Z, Hopper N (2010) Jack: Scalable accumulator-based nymble system. In: Proceedings of the 9th annual ACM workshop on Privacy in the electronic society, pp 53–62
24. Tsang PP, Kapadia A, Cornelius C, Smith SW (2011) Nymble: Blocking misbehaving users in anonymizing networks. IEEE Trans Dependable Secure Comput 8(2):256–269
25. Tsang PP, Au MH, Kapadia A, Smith SW (2007) Blacklistable anonymous credentials: blocking misbehaving users without ttps. In: Proceedings of the 14th ACM conference on Computer and communications security, pp 72–81
26. Tsang PP, Au MH, Smith SW, Kapadia A (2010) Blac: Revoking repeatedly misbehaving anonymous users without relying on ttps. ACM Trans Inf Syst Secur (TISSEC) 13(4):39
27. Cao X, Zeng X, Kou W, Hu L (2009) Identity-based anonymous remote authentication for value-added services in mobile networks. IEEE Trans Veh Technol 58(7):3508–3517
28. Liu J, Zhang Z, Chen X, Kwak K (2014) Certificateless remote anonymous authentication schemes for wireless body area networks. IEEE Trans Parallel Distrib Syst:332–342

Hongwei Li (M'12) is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, where he received the Ph.D. degree in computer software and theory, in 2008. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, for one year until 2012. His research interests include network security, applied cryptography, and trusted computing. He serves as an Associate Editor of *Peer-to-Peer Networking and Applications*, a Guest Editor of Peer to-Peer *Networking and Applications* of the Special Issue on Security and Privacy of P2P Networks in Emerging Smart City. He serves on the Technical Program Committees for many international conferences, such as the IEEE INFOCOM, the IEEE ICC, the IEEE GLOBECOM, the IEEE WCNC, the IEEE SmartGridComm, BODYNETS, and the IEEE DASC. He is also a member of the China Computer Federation and the China Association for Cryptologic Research.



Kun Jia (S'14) received the B.S. degree from the School of Computer Science and Engineering, University of Yanshan, China, in 2013. he is currently pursuing the master's degree with the School of Computer Science and Engineering at University of Electronic Science and Technology of China. He serves as a reviewer of *Peer-to-Peer Networking and Application*. His research interests include cryptography, cloud computing security, and the secure smart grid.



Haomiao Yang (M'12) received the M.S. and Ph.D. degrees in computer applied technology from the University of Electronic Science and Technology of China (UESTC) in 2004 and 2008, respectively. From 2012 to 2013, he was a Postdoctoral Fellow at Kyungil University. He is Currently an Associate Professor at the School of Computer Science and Engineering, UESTC. His research interests include cryptography, cloud security, and big data security.

Dongxiao Liu (S'14) received the B.S. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2013, where he is currently pursuing the master's degree with the School of Computer Science and Engineering. He serves as a reviewer of *Peer-to-Peer Networking and Application*. His research interests include cryptography, cloud computing security, and the secure smart grid.

Liang Zhou is a professor with the National Key Lab of Science and Technology on Communication at University of Electronic Science and Technology of China, China. His current research interests include error control coding, secure communication and cryptography.