Feng, Wei; Yan, Zheng; Xie, Haomeng

Anonymous Authentication on Trust in Pervasive Social Networking Based on Group Signature

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2017.2679980

Published: 01/01/2017

Please cite the original version:
Anonymous Authentication on Trust in Pervasive Social Networking Based on Group Signature

WEI FENG1, ZHENG YAN1,2, (Senior Member, IEEE), AND HAOMENG XIE1
1State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi’an 710071, China
2Department of Communications and Networking, Aalto University, 02150 Espoo, Finland
Corresponding author: Zheng Yan (zyan@xidian.edu.cn)

This work was supported in part by the NSFC under Grant 61672410, in part by the National Key Research and Development Program of China under Grant 2016YFB0800700, in part by the NSFC under Grant U1536202, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Program 2016ZDJC-06, in part by the 111 Project under Grant B16037 and Grant B08038, in part by the Ph.D. Grant of Chinese Educational Ministry under Grant 20150203110006, and in part by Aalto University.

ABSTRACT Pervasive social networking (PSN) supports instant social activities anywhere and at any time with the support of heterogeneous networks, where privacy preservation is a crucial issue. One of the effective methods to achieve privacy preservation is anonymous authentication on trust. However, few literatures pay attention to it. In this paper, we propose an anonymous authentication scheme based on group signature for authenticating trust levels rather than identities of nodes in order to avoid privacy leakage and guarantee secure communications in PSN. The scheme achieves secure anonymous authentication with anonymity and conditional traceability with the support of a trusted authority (TA). We also provide a mechanism to guarantee communications among nodes when TA is not available for some nodes. In addition, the utilization of batch signature verification further improves the efficiency of authenticity verification on a large number of messages. Performance analysis and evaluation further prove that the proposed scheme is effective with regard to privacy preservation, computation complexity, communication cost, flexibility, reliability, and scalability.

INDEX TERMS Privacy, trust, group signature, social networking, anonymous authentication.

I. INTRODUCTION
Pervasive Social Networking (PSN) supports instant social activities anywhere and at any time. With the popularity of smart phones and the development of heterogeneous networks organized by Mobile Ad-Hoc Networks (MANET), wireless networks, mobile Internet, and so on [1], PSN extends traditional online social networking with such specifically new properties as network carrier adaptability, social interaction ubiquity, and social service intelligence. It is an essential complement to traditional online social networking. In PSN, not only familiar people are socially connected, but also strangers are involved in social activities. PSN users can communicate with people in vicinity for various purposes, such as car sharing, urgent rescues, instant recommendations, etc.

With the new properties, PSN has shown a promising potential of wide usage. It is a perfect platform for many services. With the support of heterogeneous networks, PSN could provide services in a de-centralized way. For instance, PSN can support car-polling services such as Uber (https://www.uber.com) and Didi car-sharing in China (http://www.xiaojukeji.com). In this scenario, both riders and passengers are (some-what) anonymous, but connected with a central server. A passenger could also find a taxi or strangers nearby for a riding through PSN without direct involvement of a central server. Another typical application of PSN is to seek cooperation with surrounding people to buy some goods for a discount. It is common that in some shops, the clients could enjoy a discount only if they buy something to a certain amount, which may be beyond their real needs. In this case, one could rely on PSN to ask people in vicinity for cooperation. PSN can also be used to instantly request assistance and recommendation. All these applications can be supported by PSN in a decentralized and instant way. However, great benefits and convenience as PSN brings, it faces big challenges in security, which is highly important and worth investigating.

A. MOTIVATIONS
One of the most important issues in PSN is its security, trust and privacy [4], [5]. Considering the possibility of users to communicate with strangers nearby, they need to judge whether the communication parties are trustworthy.
Meanwhile, people show high concern on their privacy nowadays. It is obviously not wise to disclose user privacy, especially identity information to any strangers. In addition, private information leakage may also harm the safety of user properties. Therefore, it is expected to provide a mechanism to help users judge the trust of the communication parties while preserving their privacy at the same time in order to guarantee secure social networking in a pervasive way.

Introducing the concept of authenticating trust in an anonymous way can well solve this problem. Some work adopts pseudonyms to hide the real identities of users and prevent malicious tracing by frequently changing them. However, this method leads to an extra difficulty in authentication since the receiver cannot figure out the real identity of a message sender. Anonymous authentication on trust can well overcome the above issue. First, authenticating trust can help users to overcome uncertainty and make a wise decision. With the knowledge of the trust of the sender, the receiver can decide whether to trust the message and whether to communicate with the sender. Second, anonymous authentication on trust also hides user identities, thus preserve user privacy.

However, anonymous authentication on trust is not an issue that can be easily solved. First, the trust of an entity would change dynamically according to the behaviors of the entity, which makes authenticating trust more difficult than authenticating identity. It may request high communication and computation overheads due to the fact that the credentials and keys may also change with the trust value. Second, pervasive social networks are often organized by mobile devices. Considering their constrained computing and communication capacities, it is highly important to develop an effective and efficient scheme to authenticate trust.

### B. MAIN CONTRIBUTIONS
Achieving message authentication consists of two essential security checks, namely integrity check and identity check. Message authentication is helpful in resisting various security attacks, such as impersonation attacks, as well as guaranteeing a secure communication environment. An effective and efficient authentication scheme is essential for building up a practical and secure PSN environment. However, traditional authentication schemes may not fulfill the special requirements of PSN. For example, public key certificate based scheme [3], [10] requires a time-cost check in a Certificate Revocation List (CRL). However, in PSN, nodes are probably mobile devices with poor computation capacity and limited battery power. Checking CRL would occupy much computational resources and result in a high processing delay. Another challenge is the negative correlation between privacy and security [15]. And the more privacy achieved, the harder it is to provide service such as non-repudiation and accountability [16]. Yan et al. proposed a trustworthy authentication scheme in PSN that achieves anonymous authentication on both trust levels and pseudonyms [18]. Although it adopts a Trusted Authority (TA), it can guarantee communications between nodes when TA is not available with the help of trust tokens. A backup solution was provided to maintain communications when the trust token is expired and is not updated in time. However, the process of signature verification is still not efficient even with the help of batch verification.

To address the problems as described above, we propose a novel anonymous authentication scheme to authenticate trust based on group signature. We apply a TA responsible for trust management and distributing group keys to nodes according to their trust levels. Once receiving the group keys, the nodes are able to communicate with others by using their group keys to sign messages. By verifying the message signature, a receiver can verify the trust level of the signature generator. Once the group key meets its expiry time, the node turns to the TA for a new one. We utilize a trust decay function to deal with the condition where the TA is not available while the key is expired.

Specifically, the contribution of this paper can be summarized as below:

1) Our scheme is capable of authenticating trust in PSN in an anonymous way, in order to support trustworthy PSN with privacy preservation. Compared with our previous work [18], it achieves better performance.

2) We adopt group signature in PSN to provide anonymity. Besides, a revocation list is utilized to solve the revocation issue that is a challenge in terms of group signature.

3) We analyze the security and test the performance of the scheme. The result shows its efficiency and effectiveness.

The rest of the paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 introduces a system and threat model and our design goals. We describe the detailed design of the proposed scheme in Section 4, followed by security analysis and performance evaluation in Section 5. Finally, a conclusion is presented in the last section.

### II. RELATED WORK
Introducing pseudonyms into communications in PSN contributes to the protection of user identity privacy. Frequently changing pseudonyms effectively prevents malicious nodes from user tracking [2], [3], [10]. However, changing pseudonyms frequently brings overhead on authentication, since the message receiver need to verify whether a sender with a new pseudonym is still a valid user. Besides, it may lead to high communication and computation overhead due to the fact that the credentials and keys may also need to be changed frequently with the change of pseudonyms. The distribution and computation of these security credentials and certificates may generate much extra communication costs and computational overhead. Therefore, this kind of methods is not feasible to be adopted because the mobile devices in PSN are mostly resource-constrained with regard to communication capacities.

Another effective method to authenticate the validity of a communication party is to adopt a public key certificate [11]–[13]. However, it requires a node to conduct a time costly CRL check, which is not very suitable for PSN.
Some work suggested using Hash-based Message Authentication Code (HMAC) to replace time-consuming CRL check in authentication [14], [16], [17]. Wasef et al. proposed to embed a key pool in a node before network deployment [17]. Each key pool contains a certain number of key pairs that are applied to calculate a common key in order to generate a valid HMAC. A message would be considered valid only if the HMAC attached to it is generated from a correct key. When a user is revoked, unrevoked users would be capable of updating keys in their key pool with the help of a trusted authority. But the security of the scheme relies on the utilization of a Tamper Proof Module (TPM), which may not be available for every PSN node. Lin and Li proposed a cooperative message authentication scheme for Vehicular Ad Hoc Networks (VANETs) [15]. The scheme allows vehicle users to cooperatively authenticate a bunch of messages—signature pairs without the direct involvement of a TA in order to improve the efficiency of authentication and resist selfish behaviors.

Group Signature was firstly proposed by Chaum and van Heyst [6]. It enables users to sign messages on behalf of the group without revealing the identity information of the signer. Boneh et al. proposed a short group signature [7], with signature length under 200 bytes. Based on the short group signature, Wasef et al. presented a group signature scheme that supports batch verification [8] to improve signature verification efficiency. Based on this work, Zhu et al. designed a group signature based scheme for VANET [16]. To address the problem of key revocation, each node is issued a common master key to generate a HMAC value to validate signature. The master key is protected in TPM and is updated with a broadcasting message issued by the trusted authority and only un-revoked users can extract the secret value used to update the master key from the message. The scheme is efficient and achieves anonymity. But it does not consider the trust issue, which is very important to build practical and secure PSN.

Currently, few studies pay attention to the authentication on trust, especially in an anonymous way. We proposed a trustworthy authentication scheme, which achieves anonymity, unlinkability and traceability [18]. However, the scheme is not efficient enough. Even with batch signature verification, the cost of signature verification is high. How to authenticate trust in an effective and efficient way is still an open problem.

III. SYSTEM MODEL AND DESIGN GOALS

A. SYSTEM MODEL

PSN system consists of two kinds of entities, namely nodes and TA. The nodes are the PSN participants interacting with each other for social activities. Generally, the nodes are played by mobile devices held by users, such as mobile phones and pads, which have a relatively low computing capacity and is constrained in battery power. They could be malicious, and may collude with each other for malicious purposes. The TA is a fully trusted entity with powerful computation capacity and sufficient resources. We assume that it cannot be compromised. It can collect sufficient information to conduct accurate evaluation on node trust. To reduce computation burden, nodes may resort to TA through the mobile Internet to manage identities, issue keys, and evaluate trust for the purpose of secure communications. Notably, the nodes may not be able to always connect to the TA. Although the TA helps nodes authenticate the trust of other nodes, it is not directly involved into the communications among them. With a valid key issued by TA, nodes can authenticate each other without the presence of TA. The nodes and TA exchange messages through a secure channel, which is protected by applying some secure protocol. Therefore, attackers cannot get any information from the communications between nodes and TA.

Fig. 1 shows the system model of PSN. Each PSN node consists a trust evaluator to estimate the trust level of other nodes. Nodes could be connected by various types of networks, such as MANET, WiFi, mobile cellular networks, and so on. Different from traditional social networking, the nodes in PSN may not know with each other. Hence, PSN nodes usually face a problem that it is hard to judge whether it is secure to communicate with other nodes since a PSN node may be selfish, dishonest or even malicious. The TA is considered to be powerful and secure as aforementioned, and can collect the trust estimation results of nodes and sufficient information about node behaviors to perform trust evaluation with high accuracy. TA is also responsible for identity management with a node identity management module, and generates group private keys for nodes according to their trust level with a key issuer. With a group private key, nodes can generate the signature on messages and communicate with each other through authentication.

B. DESIGN GOALS

To achieve anonymous authentication on trust in PSN, our design should achieve the following security and performance goals: 1) privacy preservation and anonymous authentication
on trust values; 2) anonymity and unlinkability with regard to node identification and recognition; 3) conditional traceability in case disputes; 4) low computational complexity and overhead; 5) scalability to support large scale PSN and flexibility to handle various PSN scenarios.

IV. PROPOSED SCHEME

A. PRELIMINARY

1) BILINEAR PAIRING

Let $G_1$ and $G_2$ denote additive cyclic groups, and $G_T$ denote a multiplicative cyclic group of the prime order $p$. Let $g_1$ be a generator of $G_1$, $g_2$ be a generator of $G_2$, and $\psi$ be an isomorphic from $G_2$ to $G_1$. $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear map, which satisfies the following:
- Bilinear: $e(u^a, v^b) = e(u, v)^{ab}$ for all $u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z}_p$.
- Non-degeneracy: $e(g_1, g_2) \neq 1_{G_T}$.
- Admissible: $e$ and isomorphism $\psi$ are efficiently computable.

**TABLE 1. Notations.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td>$N_i$</td>
<td>The $i$th node;</td>
</tr>
<tr>
<td>$pid_{N_i}$</td>
<td>The pseudonym of $N_i$;</td>
</tr>
<tr>
<td>$H(\cdot)$</td>
<td>The hash function like SHA-1;</td>
</tr>
<tr>
<td>$SK_{TA}$</td>
<td>The key pairs of TA;</td>
</tr>
<tr>
<td>$PK_{TA}$</td>
<td>The public key pairs of TA;</td>
</tr>
<tr>
<td>$SK_{N_i}, PK_{N_i}$</td>
<td>The long-term key pairs of $N_i$;</td>
</tr>
<tr>
<td>$Cert_{N_i}$</td>
<td>The certificate of $N_i$ issues by TA;</td>
</tr>
<tr>
<td>$TV_{N_i}$</td>
<td>The trust level of $N_i$.</td>
</tr>
<tr>
<td>$GSK_{N_i}$</td>
<td>The trust based key of $N_i$</td>
</tr>
<tr>
<td>$f_{Decays}(\cdot)$</td>
<td>The decaying function of trust</td>
</tr>
</tbody>
</table>

B. SCHEME

Here, we describe our scheme with the following processes: SystemSetup, NodeRegistration, KeyIssue, SignatureGeneration, Verification, BatchVerification, and RULIssue. Table 1 summarizes the notations used in the scheme.

1) SystemSetup

In the process of SystemSetup, TA initializes the system parameters applied in the PSN system. It also establishes its own public/private key pair and other secret parameters, as shown in Algorithm 1.

Herein, $SK_{TA}, s_1, s_2, \gamma_1, \gamma_2$ should be kept secret by the TA, and $PK_{TA}, u, v, h, \lambda, \omega_1, \omega_2, g_1, g_2, G_1, G_2$ are public to all PSN nodes including the malicious nodes and attackers.

2) NodeRegistration

A node needs to register to the TA first before it joins the PSN. This process can be conducted both online and offline.

Each node sends its real unique identifier $ID_{N_i}$, and TA can verify the validity of the node. If valid, the TA will issue a certificate and long-term public/private key pair for the communication between TA and this node. The certificate and long-term public/private key pair are used to request trust based group key with TA. They are not applied in data transmission with nodes. The detailed description is as follows.

First, TA chooses a random in $\mathbb{Z}_p$ as the node’s private key $SK_{N_i}$, and calculates the public key $PK_{N_i} = SK_{N_i}g_2$. TA further signs the long-term public key with its own private key $SK_{TA}$ and takes the result as the certificate $Cert_{N_i}$. TA then assembles the key pairs and the certificate as well as other public system parameters together and generates a signature on it, sends back the key pair as well as the certificate to the nodes. Since the communication between TA and nodes is assumed to be conducted via a secure channel, the sensitive message conveyed between them is considered unable to be captured by attackers.

3) TRUST BASED KEY ISSUE AND REVOCATION

To interact with each other, nodes need to be issued a trust-based key to sign messages. Considering the dynamic properties of trust, the trust-based key should be updated periodically. When the trust-based key of a node is expired or leaked to adversaries, the node needs to request the TA for a new one. First, the node sends a request to TA, which includes its certificate, a time stamp and a signature on it. When TA receives the request, it will verify the certificate as well as the signature. After the success of verification, it generates a new trust-based key according to the trust value of the node using Algorithm 2. The trust-based group key ($GSK_{N_i}$) together with the trust value of the node, its expiry time and a new pseudonym of the node are transmitted to the node through a secure channel, which is protected by some existing secure protocols.
Algorithm 2 Trust-Based Key Generation

1. Select \( x_{N_i} \in \mathbb{Z}_p \);
2. TA sets the trust value \( TV_{N_i} \) and its expiry time \( T_{expire} \), and a random in \( \mathbb{Z}_p \) as the node’s pseudonym \( pid_{N_i} \) for the node, calculates:

\[
A_{N_i} = \left( x_{N_i} + H \left( TV_{N_i} || T_{expire_{N_i}} || pid_{N_i} \right) \right) \gamma_1 + \left( H \left( TV_{N_i} || T_{expire_{N_i}} || pid_{N_i} \right) \right)^{-1} \gamma_2 \right) \g_1
\]
3. Output \( GSK_{N_i} = (x_{N_i}, A_{N_i}) \)

4) SIGNATURE GENERATION

On receiving the trust-based group key, the node first processes the key as below for gaining efficiency:
- Calculate \( \hat{\omega} = H \left( TV_{N_i} || T_{expire_{N_i}} \right) \omega_1 + H \left( TV_{N_i} || T_{expire_{N_i}} \right)^{-1} \omega_2 \) and store the result.
- Calculate \( e(h, \hat{\omega}) \) and store the result.
- Calculate \( e(h, g_2) \) and store the result.
- Calculate \( e(\mu, g_2) \) and store the result.
- Calculate \( e(A_{N_i}, g_2) \) and store the result.
- Calculate \( e(A_{N_i}, g_2) \) and store the result.

All the above computation results can be reused to simplify the process of signature generation, among which, \( \hat{\omega}, e(h, \hat{\omega}), e(A_{N_i}, g_2) \) need to be calculated for group key update at each time, while \( e(h, g_2), e(\mu, g_2) \) need to be conducted only once. Then, the node can generate a signature for a message. The concrete algorithm is shown in Algorithm 3. After calculation of signature \( \sigma \), the node could send the message with the format \( \{M || TV_{N_i} || T_{stamp} || T_{expire_{N_i}} || pid_{N_i} \} || \sigma \}. \ T_{expire_{N_i}}, \ pid_{N_i}, \ TV_{N_i} \) only need to be conveyed in the first interaction with another node.

5) VERIFICATION AND BATCH VERIFICATION

When a node receives a message, it first extracts the pseudonym and checks whether it is in the revoked user list. If yes, the message would be discarded. If not, the receiver further checks whether the key is expired by comparing with \( T_{expire} \). Generally, expired keys should be treated as invalid. However, considering that TA may not be always available and the node may not always be able to update their keys in time, we need to offer a backup solution for the nodes to handle this situation. In our scheme, if a node cannot connect to the TA but its key is expired, it can still use its own key to generate keys. But the receiver would not treat it as a node whose trust value is what it claims. It calculates a current “valid trust value” through a designed function \( f_{vconvert}(TV, T_{expire}, T_{current}) \). The function is established by the TA and takes claimed trust value \( TV \), expiry time \( T_{expire} \) and current time \( T_{current} \) as input. The concrete implementation of the function could be various with the application scenarios, but the calculated valid trust value must be smaller than the claimed trust to guarantee the security. Here, we designed the function as:

\[
f_{vconvert}(TV, T_{expire}, T_{current}) = \frac{TV}{2^{[T_{current} - T_{expire}]/T_{valid}}}
\]
where \( T_{valid} \) is the length of valid period for a key. In this way, the node could still be involved in the social activities when the TA is not available. The receiver examines whether the sender fulfills its trust strategy and decides whether it needs to verify the signature. The signature is verified with Algorithm 4. Here, for simplicity, we denote \( H(TV_{N_i} || T_{expire_{N_i}} || pid_{N_i}) \) by \( H_{N_i} \).

To improve the efficiency, the receiver could aggregate many signatures together and verify them once using the batch verification. The concrete algorithm is described in Algorithm 5.

6) REVOCATION

The trust-based key can reveal the trust information of a node. Considering the dynamic changes of trust values, we set expiry time for the key, and when the key is expired, node must request the TA for a new one to maintain its regular communications. In the process of key updating, TA need not generate a new \( x_{N_i} \) for the node. Instead, only a new \( A_{N_i} \) is generated according its new trust level, new expiry time and a new pseudonym. However, there exists a problem that the signature verification, which will be introduced in the following part, could not filter a signature that is generated by
Algorithm 4 Verification

Require: msg, g₁, g₂, u, v, h, ω₁, ω₂
1. Set: \( \tilde{R}_1 = -cT_1 + s_uu \)
2. Set \( \tilde{R}_2 = -cT_2 + s_vv \)
3. Set \( \tilde{R}_3 = s_uT_1 - s_uu \)
4. Set \( \tilde{R}_5 = s_vT_2 - s_vv \)
5. Set
   \[
   \tilde{R}_5 = e(s_uT_3, g_2)e(\beta H_NH_3 - (s_u + s_v)H_Nh, \omega_1) \\
   \times e(h, g_2)^{-s_u} \\
   \times e(g_1, g_2)^{-c}
   \]
6. If \( c = (\tilde{R}_3 H(M||T_{stamp})+T_1+T_2+T_3+\tilde{R}_1+\tilde{R}_2+\tilde{R}_4+\tilde{R}_5) \mod p \)
   Then the signature is valid. 
else
   Reject the signature.

Algorithm 5 Batch Verification

Require: msg₁, msg₂, . . . . . , msgₙ, g₁, g₂, u, v, h, ω
1. Set \( \sum_{i=1}^{n} \tilde{R}_{1,i} = -\sum_{i=1}^{n} c_i T_{1,i} + \sum_{i=1}^{n} s_{ui}u \)
2. Set \( \sum_{i=1}^{n} \tilde{R}_{2,i} = -\sum_{i=1}^{n} c_i T_{2,i} + \sum_{i=1}^{n} s_{vi}v \)
3. Set \( \sum_{i=1}^{n} \tilde{R}_{3,i} = -\sum_{i=1}^{n} s_{ui}T_{1,i} + \sum_{i=1}^{n} s_{si}u \)
4. Set \( \sum_{i=1}^{n} \tilde{R}_{4,i} = -\sum_{i=1}^{n} s_{vi}T_{2,i} + \sum_{i=1}^{n} s_{si}v \)
5. Set \( \sum_{i=1}^{n} \tilde{R}_{5,i} = e(\sum_{i=1}^{n} s_{ui}T_{3,i}, g_2)e(c_i H_N T_{3,i} + \sum_{i=1}^{n} s_{ui}u \cdot \sum_{i=1}^{n} s_{vi}v, \omega_1) (\sum_{i=1}^{n} c_i, g_1, g_2) \)
6. If \( \sum_{i=1}^{n} c_i \mod p = \sum_{i=1}^{n} (\sum_{i=1}^{n} \tilde{R}_{5,i}) \mod p \)
    then
    7. Accept the signatures
   8. else
    9. Reject the signatures
10. End if

A revoked group key that is still in its valid period. This situation may happen when a valid key is leaked to attackers or a node performs some malicious activities leading to key expiry before its expiry time. To address this problem, the TA needs to issue a revoked user list that contains a list of the pseudonyms of revoked users with unexpired keys. When performing signature verification, the receiver should first check whether the message sender is included in the list. If yes, the signature should be seen as invalid and the verification fails. We must note that, only the pseudonyms of users, whose keys are revoked due to some reason but still within their valid period, should be contained in the list. When these keys reach their expiry time, the related pseudonyms should be removed from the list. Therefore, the length of the list does not grow up linearly with time, but remains reasonable. This means that checking the revoked user list would not generate much computation cost. When updating the list, the TA just needs to issue the appending index of the pseudonym rather than the whole list. After receiving the updating message, the node adds the pseudonyms in the newly received list to the revoked user list maintained by itself, and removes the pseudonyms of users whose related trust based keys have reached their expiry time. Therefore, the communication cost of updating the list is reasonable.

V. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

A. SECURITY ANALYSIS

In this part, we analyze the security properties of our scheme, namely correctness, anonymity, unlinkability, conditional traceability and nonrepudiation.

1) CORRECTNESS

Definition 1: The process of authentication is correct if a correctly generated signature always passes signature verification.

Theorem 1: The proposed scheme satisfies authentication correctness.

FIGURE 2. Authentication procedure.
By

From the result (2) and (4), we can safely get that:

Substitute $s_d = r_u + c\alpha$, $s_p = r_\beta + c\beta$, $s_x = r_x + cs_N, s_\delta = r_\delta + s_\delta$, $s_\mu = r_\mu + c\mu$ into (1) and by some simple deduction, we have that:

Denote

So we can get:

Since $\delta = \alpha x_N$, and $\mu = \beta x_N$, we can have:

Therefore, we could get the following equation:

Denote $H(T_{\text{expire}_N}||TV_N)\gamma_1 + H(T_{\text{expire}_N}||TV_N)^{-1}\gamma_2$ by $H_N$

Then we have:

We substitute $\delta = \alpha x_N$, and $\mu = \beta x_N$ into (3), we can get:

From the result (2) and (4), we can safely get that:

That is, the signature can pass the verification.

For batch verification, we can have

Then, the signature generated with a correct private key can pass the verification.

2) ANONYMITY

Definition 2: If the process of authentication cannot reveal any information of real identity of a node, the scheme fulfills anonymity.

Proposition 2: Our scheme achieves anonymity.

Proof: The real identity of node $N_i$ is preserved within the TA. The trust-based group key to sign a message is generated from the trust level and pseudonym of a node, which has no trace of the real identity. All the inputs to generate a valid signature are not related to the real identity of nodes. Besides, the pseudonym of a node changes every time it updates its trust level. Therefore, the scheme satisfies anonymity.

3) UNLINKABILITY

Proposition 2: Our scheme achieves unlinkability when using different group private keys.

Proof: In our scheme, when transmitting a message, some parameters such as pseudonym, trust value and expiry time may be utilized by attackers to judge whether two messages are from a same node. However, in a session with the same node, the sender only needs to send the above parameters in the first message. After that, the receiver would store them. Therefore, the attacker could only take advantage of the content of the message as well as the signature to judge whether two messages are from the same person. However, the group signature scheme we adopt is proved to fulfill unlinkability [8]. Although we embedded the trust information in it, the attacker cannot extract any information about it. Thus, our scheme fulfills unlinkability.

4) CONDITIONAL TRACEABILITY & NON-REPUTATION

Proposition 3: Our scheme achieves conditional traceability and non-reputation. Only the TA can trace a node with the knowledge $s_1$ and $s_2$. When a dispute happens, TA will first get the signature causing the dispute. TA will first verify the signature and see whether it is a valid signature. If yes, it will identify the node that has sent the message as follows:

- Extract $T_1, T_2, T_3$ from the signature;
- Get the part of the private key of the sender by calculating $A = T_3 - s_2 T_1 - s_1 T_2$.
- Find the identifier attached to the $A$.

In this way, TA can trace the node. However, since no parities except TA has the knowledge of $s_1$ and $s_2$, others cannot trace the identity of a message signer.
B. PERFORMANCE ANALYSIS

In this section, we evaluate the performance of our scheme in terms of computation complexity, communication cost, scalability, and flexibility.

1) COMPUTATION COMPLEXITY

We only consider the time consuming algorithms when analyzing the computation complexity of our scheme. The main computation cost is caused by bilinear pairing and exponentiation operations in $G_1$, $G_2$, and $G_T$.

Since the algorithm SystemSetup consists of constant number of operations and is only performed once by TA during the lifetime of PSN, the computation complexity of SystemSetup is $O(1)$.

NodeRegistration is conducted for every node that joins PSN, supposing the number of nodes in PSN system is $N$, its computation complexity is $O(N)$.

The algorithm KeyIssue, which is performed by TA, contains only one exponentiation operation in $G_1$, but it needs to be conducted every time a node requests a new group key. Therefore, its computation complexity is $O(n)$, where $n$ the number of nodes that request a new group key is.

The most costly algorithms are SignatureGeneration and Verification. For signature generation, since some of the calculated results could be reused, it takes eight exponentiation operations in $G_1$ and five exponentiation operations in $G_T$ excluding the operations performed before signature generation. But for signature verification, even though some computing results can be pre-calculated and stored, a node still needs to perform thirteen exponentiation operations in $G_1$, three bilinear pairing and two exponentiation operations in $G_T$ to verify a signature. The computation complexity is $O(m)$, where $m$ is the number of messages transmitted in PSN. To improve the efficiency, batch signature verification is supported. By aggregating many signatures and verify them at one time, our scheme achieves improved performance. In the worst case where a node receives $m$ messages from $m$ different nodes, the node needs to perform $7n + 6$ exponentiation operations in $G_1$, two exponentiation operations in $G_T$ and only three pairing operations. The number of pairing operations is a constant. Although the complexity is still $O(m)$, the number of operations is much reduced.

Table 2 summarizes the computation complexity of each system operation in our proposed scheme. The performance of our scheme is mainly influenced by the number of nodes and the number of messages transmitted in PSN.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Computation Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemSetup</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>NodeRegistration</td>
<td>$O(N)$</td>
</tr>
<tr>
<td>GroupKeyIssue</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>SignatureGeneration</td>
<td>$O(m)$</td>
</tr>
<tr>
<td>Verification</td>
<td>$O(m)$</td>
</tr>
<tr>
<td>BatchVerification</td>
<td>$O(m)$</td>
</tr>
</tbody>
</table>

$N$ is the total number of nodes in PSN; $n$ is the number of nodes requesting new group keys; $m$ is the number of messages transmitted in PSN.

2) COMMUNICATION COST

The communication cost of our proposed scheme mainly consists of three parts after system setup and node registration: the broadcast of an aggregated pseudonym list, group private key issue and message exchange. The aggregated pseudonym list is composed of two parts, namely the pseudonyms of the revoked nodes and the signature of TA on the list. We should note that only the malicious nodes whose keys are not expired would be involved in the list. Once the keys reach their expiry time, the pseudonyms of them would be removed from the list. Therefore, the length of the revoked pseudonym list would be short. The signature of the list is an element of $G_1$. Therefore, in the process of issuing the list of revoked pseudonyms, the communication is reasonable.

The communication cost of the proposed scheme mainly consists of two parts after system setup and node registration: the issue of group key and the message exchange. The issued group key contains one element in $G_1$ and one element in $Z_p$. Since in our implementation, the size of element in $G_1$ is 40 bytes and the size of element in $Z_p$ is 20 bytes, the communication cost of group key issue is 60 bytes, which is reasonable. The frame of message is $|M| + |TV_N| + |T_{stamp}| + |T_{expire}| + |pid_N| + |\sigma|$, in which $M$ is the message transmitted and cannot be avoided in any schemes. The signature includes six elements in $Z_p$ and three elements in $G_1$, since the size of elements in both $Z_p$ and $G_1$ are short (20 bytes for an element in $Z_p$ and 40 bytes for an element in $G_1$), the size of the signature is reasonable. The size of trust value and expiry time, and the pseudonym are all are set as two bytes, which has little influence on the communication cost. The total size of a message is 248 bytes, which is reasonable.

3) SCALABILITY

In our scheme, a certificate is not required in the communications among nodes. The trust level and the validity of a message sender can be authenticated by verifying the group signature generated by the sender. We adopt a revoked user list to support the validation of the message sender in the procedure of verification, which has been illustrated that its length is reasonable and the check does not cause much computation overhead. Besides, the computation overhead could be greatly reduced by using batch verification. Since the public key certificate based schemes usually suffer from time-consuming CRL check, excluding certificate could dramatically reduce verification overhead. Thus this design improves the scalability of the system to some extent.

4) RELIABILITY AND FLEXIBILITY

The proposed scheme takes advantage of TA to provide a reliable authentication mechanism by applying the TA to
undertake some operation tasks in order to release the computation burden of the nodes. However, TA is not directly involved in the communication between nodes. With the group key issued by TA, a node can still communicate with others. Besides, we offer a solution to enable a node to maintain common communications even the group key meets expiry time and the TA is not available by adopting a trust decay function. The function can be decided by the TA according to their strategy flexibly.

### C. SCHEME OPERATION PERFORMANCE

We implemented the proposed scheme in C++ language using a PBC library for algebraic operations. The scheme was implemented on a laptop running 64-bit Ubuntu Linux 14.04 with 2.5 GHz Intel Core i5 Quad-CPU and 4.0G RAM. Table 3 shows the average execution time of each basic algorithm.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Operation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemSetup</td>
<td>18.27 ms</td>
</tr>
<tr>
<td>NodeRegistration</td>
<td>6.99 ms</td>
</tr>
<tr>
<td>GroupKeyIssue</td>
<td>1.39 ms</td>
</tr>
<tr>
<td>SignatureGeneration(with Pre-computation)</td>
<td>17.04 ms</td>
</tr>
<tr>
<td>Verification</td>
<td>27.05 ms</td>
</tr>
<tr>
<td>BatchVerification (Average for M=1500)</td>
<td>4.978 ms</td>
</tr>
</tbody>
</table>

Fig. 3 shows the operation time of system setup, node registration, and group key issue. Node Registration is increased linearly with the number of nodes registered in PSN, since it is needed to be conducted for every node in PSN. The main cost is caused by one exponentiation operation in $G_2$, which costs about 5.51 ms. The implementation results conform to our analysis.

Fig. 4 shows the computation cost of signature generation, which increases linearly with the number of messages to be signed. We could observe that by computing some of the variables ahead of signature generation, which could be avoided in the later signature generation as long as the group private key does not change, the operation time could be much reduced. The average operation time to generate a signature with pre-computation is 17.04 ms.

The operation time of verification and batch verification is shown in Fig. 5. The verification is costly (27.05 ms for each signature). However, with batch verification, most of the time-cost operations could be avoided, thus the efficiency is greatly improved. When verifying 1500 signatures, the average verification time is only 4.975 ms.

### D. COMPARISON

We further compared our scheme with our previous work [18] to show the improvement of the proposed scheme. First, we compared the performance of key generation, signature generation and verification of the two schemes. The comparison result shown in Table 4 indicates that the newly proposed scheme is better than our past work in terms of group key generation and batch signature verification. The signing key generation in our previous work costs much more than the
TABLE 4. Comparison result based on operation time.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Current Scheme</th>
<th>Previous Scheme [18]</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyIssue</td>
<td>1.39ms</td>
<td>14.42ms</td>
</tr>
<tr>
<td>SignatureGeneration</td>
<td>17.04ms</td>
<td>10.83ms</td>
</tr>
<tr>
<td>BatchVerification</td>
<td>4.978ms</td>
<td>23.86ms</td>
</tr>
</tbody>
</table>

newly proposed one, and it is performed by the node itself. While in the newly scheme, the signing key generation takes only 1.39ms and is performed by the TA. To generate a valid signature, the new scheme costs a bit more time than the previous one and falls into the same cost level, but the signature verification is much more efficient than the previous one (only 4.978ms is needed). Therefore, the scheme presented in this paper is more efficient than the previous one described in [18], especially for signing key generation and signature verification. It is more proper to be applied into the scenarios that key generation and user revocation are frequently needed and is good for verifying a big number of messages.

TABLE 5. Comparison result based on communication cost.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Current Scheme (byte)</th>
<th>Previous Scheme [18] (byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Registration</td>
<td>260</td>
<td>276</td>
</tr>
<tr>
<td>Signing Key Issue</td>
<td>60</td>
<td>148</td>
</tr>
<tr>
<td>Message Transmission</td>
<td>248</td>
<td>276</td>
</tr>
<tr>
<td>List Issue</td>
<td>40+20n</td>
<td>128+20N</td>
</tr>
</tbody>
</table>

n is the number of revoked nodes with a valid key pair, N is the total number of nodes in PSN, and n is much smaller than N.

The scheme proposed in this work is also superior to the previous one in terms of communication cost. We compared the communication cost of the two schemes and the comparison result is illustrated in Table 5. We compared them in four procedures, namely node registration, signing key issue, message transmission and revocation user list/aggregate token list issue. The comparison result shows the fact that in all the four procedures, the newly proposed method is less costly than the previous one. For the signing key issue, the new scheme only needs to send a message of 60 bytes to the nodes, while the previous scheme needs 148 bytes. The revocation user list for the purpose of user revocation in the new method is much shorter than the aggregate token list in the previous one. This is mainly due to the fact that, in the previous work, the list is linear to the number of nodes in the whole PSN network. While in current work, it is only linear to the number of revoked users whose keys have not reached their expiry time, which is much smaller than the total number of users.

We additionally compared the key security features of the two schemes, namely signing key issue, revocation, privacy preserving, unlinkability, and resisting potential attacks. Both schemes issue signing keys through a secure channel, and provide an extra solution for the case where the TA is not available. However, in our scheme, with the pre-defined decaying function, no extra keys are needed to guarantee the communication. Our scheme enables TA to define various decaying functions, thus more flexible. Based on pseudonym, our scheme can protect the identity of the node securely. Besides, as analyzed in Section 5.1, our scheme can achieve unlinkability within the same session. While in [18], if a node signs messages with different private keys generated by the same token, the attackers could decide that the two messages are sent by the same node. Our scheme thus achieves better unlinkability. At last, both schemes could resist potential attacks, such as replay attack and impersonate attack. Table 6 summarizes our comparison results.

We also compared our scheme with the scheme proposed in [16], which adopts the same group signature scheme. We revise the group signature scheme to make it capable of verifying the trust level of the signer. The modification does not introduce much extra computation overhead. There is no much difference in terms of computation efficiency between the two schemes. However, our scheme can offer several advantages. First, our scheme supports verification on trust, which is not supported by the work in [16]. Second, we provide a more efficient method for issuing the revocation list than [16]. As aforementioned, only the malicious nodes whose group private keys are still within the valid period should be contained in the list that makes the length of the revocation list reasonable. If a malicious node’s key has reached its expiry time, the TA could just stop issuing new keys for it to revoke the node. If the key of the malicious node is still within its valid period, the TA just needs to add the pseudonym of the node into the list and issue the list. While in [16], to revoke a malicious node, each valid node needs to compute a polynomial. When the number of user is big, this approach suffers from heavy computation overhead, which could not be afforded by PSN nodes with limited resources. Third, unlike the work in [16], the security of our scheme does not rely on the usage of TPM.

TABLE 6. Comparison result based on main features.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Comparison result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signing Key Issue</td>
<td>Both can issue keys securely, but the scheme proposed in this paper is more efficient and flexible.</td>
</tr>
<tr>
<td>Unlinkability</td>
<td>The proposed scheme can support unlinkability within a same session.</td>
</tr>
<tr>
<td>User Revocation</td>
<td>The proposed scheme supports user revocation more efficiently.</td>
</tr>
<tr>
<td>Privacy Preservation</td>
<td>Both can preserve identity privacy.</td>
</tr>
<tr>
<td>Resistance to Attacks</td>
<td>Both can resist replay attack and impersonate attack.</td>
</tr>
</tbody>
</table>

VOLUME 5, 2017
VI. CONCLUSION

In this paper, we proposed an anonymous authentication scheme based on group signature for authenticating both pseudonyms and trust levels in order to support trustworthy PSN with privacy preservation. The scheme achieves secure anonymous authentication with the support of TA for conditional traceability. Although the scheme applies a centralized trusted authority, the TA is not necessary to be always available when exchanging messages between nodes. Even when the group key cannot be updated in time, the node could continue to communicate with others with the help of the trust decay function based on a certain trust strategy, which is very flexible. The utilization of batch verification further reduces the computation cost of the node for signature verification. The utilization of revocation list can efficiently prevent malicious nodes from participating in PSN activities due to its reasonable length. The performance analysis and scheme implementation further showed the efficiency and effectiveness of the scheme.

REFERENCES


WEI FENG received the B.Sc. degree in telecommunications engineering from Xi’an Jiaotong University, Xi’an, China, in 2014, where he is currently pursuing the Ph.D. degree in information security. His research interests include information security, privacy preservation, and trust management in social networking and mobile crowdsourcing.

ZHENG YAN (M’06–SM’14) received the B.Eng. degree in electrical engineering and the M.Eng. degree in computer science and engineering from Xi’an Jiaotong University, Xi’an, China, in 1994 and 1997, respectively, the M.Eng. degree in information security from the National University of Singapore, Singapore, in 2000, and the Licentiate of Science and Doctor of Science degrees in technology in electrical engineering from the Helsinki University of Technology, Helsinki, Finland, in 2005 and 2007, respectively. She is currently a Professor with the Xi’an University and also a Visiting Professor with Aalto University, Espoo, Finland. She has authored over 150 publications and solely authored two books. She is an Inventor of over 50 patents and patent applications. Her research interests are in trust, security and privacy, social networking, cloud computing, networking systems, and data mining. She serves as an Organization and Program Committee Member for numerous international conferences and workshops. She is also an Editor or a Guest Editor of many reputable journals, such as, the Information Sciences, the ACM TOMM, the Information Fusion, the IEEE SYSTEMS JOURNAL, the IEEE ACCESS, and the IEEE IOT JOURNAL.

HAOMENG XIE received the B.Sc. degree in telecommunications engineering from Xi’an Jiaotong University, Xi’an, China, in 2016, where he is currently pursuing the master’s degree in information security. His research interests are in security, privacy preservation, and trust management in social networking.

***