Universal forgery on a group signature scheme using self-certified public keys

Guilin Wang

Infocomm Security Department, Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613

Received 19 March 2003; received in revised form 25 November 2003
Communicated by K. Iwama

Abstract

A group signature scheme allows any group member to sign messages on behalf of the group in an anonymous and unlinkable fashion. In the event of a dispute, a designated group manager can reveal the identity of the signer. In 1999, Tseng and Jan proposed a group signature scheme using self-certified public keys. By attacking their signature verification equation, we demonstrate that their scheme is universally forgeable, i.e., anybody can forge a valid group signature on any message such that the group manager is unable to determine the identity of the signer.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Cryptography; Digital signature; Group signature; Forgery

1. Introduction

Group signatures are first introduced and realized by Chaum and van Heyst in [5]. In a group signature scheme, each group member of a given group is able to sign messages anonymously on behalf of the group. By using the single group public key, a verifier can check the validity of a group signature but he is unable to find who has signed it. However, in case of later dispute, the group manager can open a group signature and then identify the true signer of it. In virtue of these advantages, group signatures are especially useful in many practical applications, such as anonymous authentication, e-voting, e-bidding and e-cash, etc. Generally speaking, a secure group signature scheme must satisfy the following properties [4,3]:

- Unforgeability: Only group members are able to sign messages on behalf of the group.
- Anonymity: Given a valid signature of some message, identifying the actual signer is computationally hard for everyone but the group manager.
- Unlinkability: Deciding whether two different valid signatures were computed by the same group member is computationally hard.
- Exculpability: Neither a group member nor the group manager can sign on behalf of other group members.
- Traceability: The group manager is always able to open a valid signature and identify the actual signer.
Coalition-resistance: A colluding subset of group members (even if comprised of the entire group) cannot generate a valid signature that the group manager cannot link to one of the colluding group members.

In 1999, Tseng and Jan proposed a group signature scheme based on the notions of self-certified public keys [6] and identity-based cryptosystems [7]. Later, Ateniese, Joye and Tsudik pointed out that Tseng–Jan scheme is not coalition-resistant because two colluding members can generate a new membership certificate [2].

In this paper, however, we show that Tseng–Jan scheme [8] is universally forgeable, i.e., anyone (not necessarily a group member) is able to easily generate a valid group signature on any message such that even the group manager is unable to open it. Moreover, our attack just uses the signature verification equation as the target. This means that our attack can also be applied on any improvement of Tseng–Jan scheme no matter what changes are made in the membership certificate generation and signature equation as long as the verification equation is the same. At the same time, we not only describe how to attack this scheme, but also explain why and how we find our attack.

The rest of this paper is organized as follows. Section 2 reviews Tseng–Jan scheme [8], and Section 3 describes a conspiracy attack by Ateniese, Joye and Tsudik [2]. Then, we present our universal forgery on Tseng–Jan scheme in Section 4. Finally, conclusion is given in Section 5.

2. Tseng–Jan scheme review

In this section, we give a short description of the Tseng–Jan group signature scheme using self-certified public key and refer to the original paper [8] for more details. The scheme involves four parties: a trusted authority (TA, for short), a group manager (GM, for short), the group members, and the verifiers. The TA acts as a third party to setup the system parameters while the GM selects the group public/secret keys. Any user gets his individual secret key from the TA but uses his identity information as his individual public key [7]. He applies his group secret key from the GM only when he wishes to join the group controlled by the GM. Therefore, the GM and the TA jointly issue membership certificates to new users. In case of disputes, the GM opens the contentious group signatures to reveal the identity of the actual signer.

To setup the system, a trusted authority (TA) sets an RSA modulus \( n := pq \) with \( p := 2p' + 1 \) and \( q := 2q' + 1 \) where \( p, q, p', q' \) are all prime. Then, he selects an element \( g \in Z_p^* \) of order \( v := p'q' \) and \( e, d \in Z_v^* \) satisfying \( ed = 1 \mod v \). Furthermore, the trusted authority chooses a publicly known hash function \( f(\cdot) \), and publishes his public key \( (n, e, f(\cdot)) \), but keeps his secret key \((p, q, d)\) private.

When a group manager (GM), with identity information \( GD \), wants to establish a group, he chooses a secret key \( x \), then computes \( z := g^x \mod n \) and sends \( z \) to the TA. After receiving \( z \), the TA first evaluates \( GID := f(GD) \), then calculates and sends the following:

\[
y := z^{GID^{-1}} \mod n,
\]

\[
s_G := z^{-d} \mod n.
\]

Finally, the group manager chooses a publicly known hash function \( h(\cdot) \), publishes \((y, h(\cdot))\) as his public key but keeps \((x, s_G)\) as his secret key. Note that the GM can check the validity of his key pair by \( s_G^{e_G} = y^{-GID} \mod n \).

To join the group, a user \( U_i \), with identity information \( DI_i \), first selects his secret key \( s_i \), then computes \( z_i := g^x \mod n \) and sends \( z_i \) to the TA. The TA sends back \( p_i := (z_i)^{ID_i^{-1}} \mod n \), where \( ID_i := f(DI_i) \), and \( U_i \) checks whether \( p_i^{ID_i} \equiv z_i \mod n \). If \( p_i \) is correct, user \( U_i \) sends \( p_i \) to the GM, and the group manager returns the following value \( x_i \) to \( U_i \):

\[
x_i := p_i^{ID_i^{-1}} \cdot s_G \mod n.
\]

At last, \( U_i \) checks whether \( x_i^{e_G} \equiv y^{GID(x_i-1)} \mod n \) holds. If the answer is yes, the user \( U_i \) stores his membership certificate \((s_i, x_i)\).

To sign a message \( m \), the user \( U_i \), with his certificate \((s_i, x_i)\), first randomly selects three numbers \( r_1, r_2 \) and \( r_3 \), then computes his signature \((A, B, C, D, E)\) as follows

\[
A := r_1s_i,
B := r_2^{-A} \mod n,
C := y^{GID'A^{-}r_3} \mod n,
D := s_i \cdot h(m||A||B||C) + r_3C,
E := x_i \cdot r_2^{(h(m||A||B||C||D))} \mod n.
\]
To verify the validity of signature \((A, B, C, D, E)\) on message \(m\), a verifier checks whether

\[
y^{GID\cdot A\cdot D} \equiv (E^{e\cdot A} B^{h(m||A||B||C||D)})^{h(m||A||B||C)} \cdot C^{C} \mod n.
\]  

(4)

In case of disputes, the group manager can open a signature \((A, B, C, D, E)\) for message \(m\) by checking which \(x_t\) satisfies the following equality:

\[
(x_t)^{y^{A}} B^{h(m||A||B||C||D)} \equiv E^{e\cdot A} \mod n.
\]  

(5)

Before discussing the security of Tseng–Jan scheme, we first verify its correctness, i.e., all group signatures generated by group members satisfy the verification equation (4). For simplicity, denote \(h := h(m||A||B||C||D)\) and \(\hat{h} := h(m||A||B||C||D)\). First of all, for any membership certificate \((s_i, x_i)\), from Eq. (2) and the definitions of \(p_i, z_i, s_G\), we have

\[
x_t = p_t^{ID_i \cdot x_i} \cdot s_G = (z_i)^{d} \cdot s_G = (x_i)^{-n+1} \mod n.
\]  

(6)

Then, from two equations in (1), we have

\[
x_t = (x_t)^{-n+1} = (y^{GID\cdot d\cdot (n-1)}) \mod n.
\]

Therefore, given any group signature pair \((A, B, C, D, E)\) which is generated by user \(U_i\) according to signing Eq. (3), we have \(E^{e\cdot A} B^{h} \equiv (x_i^{e\cdot A} r_2^{\bar{e} \cdot h}) \cdot r_2^{-\bar{e} \cdot h} = x_i^{e\cdot A} = y^{GID\cdot A\cdot (n-1)} \mod n\). Hence, the following equations justify the correctness of the Tseng–Jan scheme:

\[
(E^{e\cdot A} B^{h})^{y^{GID\cdot A}} \cdot C^{C} \equiv (y^{GID\cdot A\cdot (n-1)})^{y^{GID\cdot A}} \cdot y^{GID\cdot A\cdot e} \cdot C \mod n
\]

\[
= \bar{x} \mod n
\]

\[
= y^{GID\cdot A\cdot D} \mod n.
\]

3. Atienese, Joye and Tsudik attack

In [2], Atienese, Joye and Tsudik pointed out a conspiracy attack on the above Tseng–Jan scheme such that two colluding group members can recover the group secret key \(s_G = g^{(-d)} \mod n\), and then generate a valid but illegal membership certificate. Now we describe the details. Assume that two colluding group members \(U_1\) and \(U_2\) have certificates \((s_1, x_1)\) and \((s_2, x_2)\), respectively. Let \(c := \text{gcd}(s_1 - 1, s_2 - 1)\) (Atienese et al. [2] only discussed the case of \(c = 1\)). Then, by using extended Euclidean algorithm, they can find \(\alpha, \beta \in \mathbb{Z}\) such that \(c = \alpha (s_1 - 1) + \beta (s_2 - 1)\). Hence, from Eq. (6), they can find the following value \(s_G^c\):

\[
s_G^c = s_G^{\alpha (s_1 - 1) + \beta (s_2 - 1)} = x_1^{-\alpha} x_2^{-\beta} \mod n.
\]

Consequently, using the value of \(s_G^c\), \(U_1\) and \(U_2\) can produce a new membership certificate \((\tilde{s}, \tilde{x})\) as follows: First choose a random number \(r\), then define \(\bar{x} := cr + 1\) and \(\tilde{x} := (s_G^c)^{-r} \mod n\), respectively. It is easy to know that \((\tilde{s}, \tilde{x})\) is a valid certificate because \(\bar{x} = (s_G^c)^{-r} = s_G^{1-cr+1} = s_G^{-\tilde{x}+1} \mod n\). Using such valid but illegal membership certificates, valid but untraceable group signature can be generated by using signing Eq. (3).

4. Our attack

In the above Atienese, Joye and Tsudik attack, to derive the secret value of \(s_G^c\) two colluding group members are necessary. However, as pointed in [1], in some specific situations the group members have no incentives to collude. Therefore, the emphasis is to guarantee that a single member is unable to generate valid but untraceable group signatures, instead of several colluding group members are able to generate valid but untraceable group signatures. An example situation is the Internet lottery, where each group member anonymously signs its own numbered ticket (to make it valid) and the group manager later distributes prizes to the right holders of the randomly drawn ticket numbers [2]. Therefore, it seems that Atienese, Joye and Tsudik attack does not harm to the application of Tseng–Jan scheme in such situations.

However, in our following attack, anyone (not necessarily a group member) is able to generate a valid group signature on any arbitrary message such that the group manager is unable to open it. In other words, Tseng–Jan scheme [8] is universally forgeable. Details are given bellow.

Now, we want to forge a group signature on arbitrary given message \(m\) under the assumption that we do not know any membership certificate \((s_i, x_i)\). Note that the verification equation (4) is about some powers of \(y, B, C\) and \(E\). Therefore, in order to get a quintuple \((A, B, C, D, E)\) satisfying Eq. (4), we first define \(B, C, E\) as some known powers to the base \(y\) and choose a random value for \(A\), then try to find the value \(D\) from
Eq. (4). For this sake, choose four random numbers \(\bar{a}_1, \bar{a}_2, \bar{a}_3\) and \(A\), then define \(B, C, E\) as follows:

\[
B := y^\bar{a}_1 \mod n; \quad C := y^\bar{a}_2 \mod n; \quad E := y^\bar{a}_3 \mod n.
\]

Then, from the verification equation (4), we get the condition for the value \(D\):

\[
GID \cdot A \cdot D = \left[\bar{a}_3 e A + \bar{a}_1 \cdot h(m \| A \| B \| C \| D)\right] h(m \| A \| B \| C) + GID \cdot A \cdot h(m \| A \| B \| C) + \bar{a}_2 C \mod v. \tag{7}
\]

In the above complicated equation, we do not know the value of the modulus \(v\) (it is AT’s trap-door information), and \(D\) appears in both sides and is embedded in the hash value \(h(m \| A \| B \| C \| D)\). Therefore, at the first glance, it seems unlikely to find a quintuple \((A, B, C, D, E)\) such that Eq. (7) is satisfied. However, we are attackers. So we have the freedom to choose some special values for \(\bar{a}_1, \bar{a}_2, \bar{a}_3\) and \(A\). In other words, to get a solution from Eq. (7), we can let these numbers satisfy some specific relationships. Based on this observation, it is not difficult to see that Eq. (7) will be trivially satisfied if the following two equations holds over integer ring \(\mathbb{Z}\):

\[
\begin{cases}
\bar{a}_3 e A + \bar{a}_1 \cdot h(m \| A \| B \| C \| D) = 0, \\
GID \cdot A \cdot D = GID \cdot A \cdot h(m \| A \| B \| C) + \bar{a}_2 C. \tag{8}
\end{cases}
\]

Now, it is feasible to solve equation system (8): First find the value of \(D\) from the second equation, and then get the value of \(\bar{a}_3\) from the first equation. To eliminate coefficients \(GID \cdot A\) and \(eA\), we choose two random numbers \(a_1, a_2\), and then re-define \(\bar{a}_1, \bar{a}_2, \bar{a}_3\) in \(\mathbb{Z}\), as follows:

\[
\bar{a}_1 := a_1 e A, \quad \bar{a}_2 := a_2 \cdot GID \cdot A.
\]

Then, from equation system (8), we find the following solution for Eq. (7):

\[
D = h(m \| A \| B \| C) + a_2 C \in \mathbb{Z},
\]

\[
\bar{a}_3 = -a_1 \cdot h(m \| A \| B \| C \| D) \in \mathbb{Z}.
\]

We summarize our attack as follows:

1. First select three random numbers \(a_1, a_2\) and \(A\).
2. Then define

\[
B := y^{a_1 e A} \mod n, \quad C := y^{a_2 GID \cdot A} \mod n,
\]

\[
D := h(m \| A \| B \| C) + a_2 C \in \mathbb{Z},
\]

\[
E := y^{-a_1 h(m \| A \| B \| C \| D) + a_2 GID} \mod n.
\]

3. Output \((A, B, C, D, E)\) as a group signature for message \(m\).

Now, we prove that our forgery is successful, i.e., the forged tuple \((A, B, C, D, E)\) is a valid group signature for message \(m\). For simplicity, let \(h := h(m \| A \| B \| C)\) and \(\bar{h} := h(m \| A \| B \| C \| D)\). Then, from the description of our attack, the following equations hold:

\[
\begin{align*}
E^{e A B^h GID} C^2 &= y^{-a_1 h e A B^h GID} \cdot y^{a_1 e A B^h GID} \cdot y^{a_2 GID A C + a_3 GID A D} \mod n \\
E^{GID A C + a_3 GID A D} &= y^{GID A C + a_3 GID A D} \mod n.
\end{align*}
\]

Therefore, forged \((A, B, C, D, E)\) is a valid group signature for message \(m\). Furthermore, it is easy to know that when such a forged signature is given, the group manager is unable to find any group member to take responsible for it. At the same time, according to signing Eq. (3), it is difficult to distinguish forged signatures and normal signatures generated by group members from their probability distributions.

In fact, the above attack can be generalized by choosing another new random number \(a_3\). More specifically, in the new attack, we first select four random numbers \(a_1, a_2, a_3, a_4\) and \(A\). Then, define \(B := y^{a_1 e A} \mod n, C := y^{a_2 GID \cdot A} \mod n, D := (ea_3 + 1) h(m \| A \| B \| C) + a_2 C \in \mathbb{Z}\) and \(E := y^{-a_1 h(m \| A \| B \| C \| D) + a_2 GID} \mod n\). It is easy to verify that the forged signature pair \((A, B, C, D, E)\) satisfies the verification equation (4).

5. Conclusion

In this paper, we pointed out that Tseng–Jan group signature scheme [8] is insecure. More specifically, we showed that anyone, not necessarily a group member, is able to easily generate a valid group signature on any message such that the group manager is unable to open it. In other words, the Tseng–Jan scheme is universally forgeable. Moreover, we just use the
signature verification equation as the target of our attack. Therefore, our attack can also be applied on any improvement of the Tseng–Jan scheme, in which modifications are made to the membership certificate generation and/or the signing equation but not to the verification equation. At the same time, we not only describe how to attack the Tseng–Jan scheme, but also explain why and how we find our attacks. In addition, the attacking method described in this paper can be used to analyze some other group signatures [9] and proxy signatures [10].

References