Security Analysis of Several Group Signature Schemes

Guilin Wang

Infocomm Security Department
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
glwang@i2r.a-star.edu.sg

February, 2003; Modified in September, 2003.

Abstract. At Eurocrypt'91, Chaum and van Heyst introduced the concept of group signature. In such a scheme, each group member is allowed to sign messages on behalf of a group anonymously. However, in case of later disputes, a designated group manager can open a group signature and identify the signer. In recent years, researchers have proposed a number of new group signature schemes and improvements with different levels of security. In this paper, we present a security analysis of five group signature schemes proposed in [25, 27, 18, 29, 10]. By using similar methods, we successfully identify several *universally forging attacks* on these schemes. Using our attacks, anyone (not necessarily a group member) can forge valid group signature on any message such that the forged signature cannot be opened by the group manager. The linkability of these schemes is also discussed. At the same time, we not only describe how to attack these schemes, but also explain why and how we find our attacks.

Keywords: digital signature, group signature, forgery, cryptanalysis.

1 Introduction

A group signature scheme, first introduced by Chaum and van Heyst in [7], allows each group member to sign messages on behalf of a group anonymously. However, in case of later disputes, a designated group manager can open a group signature and then identify the true signer. A secure group signature scheme must satisfy the following six properties [1, 2, 4, 7]:

- *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 general, group signature schemes can be classified into two different types: The schemes based on *signatures of knowledge* [4] and the schemes designed by *straight-forward* and *ad-hoc methods*. The schemes in [4, 5, 1, 22] belong to the first type, while the schemes proposed by [10, 11, 27, 24, 25, 18, 29] belong to the second type. Some of the first type schemes are provably secure, but all those schemes are not very efficient. For example, as one of the most efficient schemes belonging this type, the scheme in [5] sill needs about 13,000 RSA modular multiplications in generation and verification a group signature (see Section 5.6 of [5]). The second type schemes are very efficient since generation and verification of a signature only need to compute several standard signatures. However, no existing scheme of the second type has provable security.

In 1998, Lee and Chang presented an efficient group signature scheme based on the discrete logarithm [11]. Their scheme is obviously linkable since two same pieces of information are included in all group signatures generated by the same group member. To provide unlinkability, Tseng and Jan proposed an improved group signature scheme in [24]. But Sun pointed out that this improved scheme is still linkable [23]. At the same time, based on the Shamir's idea of identity(ID)-based cryptosystems [20], Tseng and Jan proposed an ID-based group signature scheme in [27]. However, Joye et. al [8, 9] showed that the schemes proposed in [11, 24, 26] all are *universally forgeable*, i.e., anyone (not necessarily a group member) is able to generate a valid group signature on any message, which cannot be opened by the GM. After that, Tseng and Jan improved their group signature schemes in [25] and [27], and Popescu presented a modification to the Tseng-Jan scheme [26] in [18]. In addition, Xian and You [29] proposed new group signature scheme with strong separability such that the group manager can be split into a membership manager and a revocation manager.

In this paper, we present a security analysis of several group signature schemes proposed in [25, 27, 18, 29, 10]. By using similar methods originated from [3, 8, 9], we successfully identify different *universally forging attacks* on these schemes. Using our attacks, anybody can easily forge valid group signature on an arbitrary message. At the same time, we point out that the schemes proposed in [26, 27, 18, 29] are *linkable*. In our description, we not only describe how to attack these schemes, but also explain why and how we find our attacks. On the one hand, our attacks show that the schemes mentioned above are *insecure*. On the other hand, our attacks also implied that constructing group signatures by the ad-hoc methods should be terminated. In other words, from the contrary side of the same problem, the design methodology employed in [1, 22] are confirmed.

In addition, using our method, the existing attacks on Kim et al.'s convertible group signature scheme [10] can be unified in a family. Those existing attacks are pointed out by [12, 19, 28, 6] independently and accidentally. Furthermore, we find a new problem in Kim et al.'s scheme, that is, a valid group signature signed by one group member is also a possible valid group signature of other group members for the same message. Therefore, their group signature scheme is information-theoretically *anonymous* even for the group manager, and hence all valid group signatures are completely *untraceable* and *unlinkable*.

The rest of this paper is organized as follows. We review and analyze Tseng-Jan scheme I [25], Tseng-Jan scheme II [27], Popsecu's scheme [18], Xia-You scheme [29], and Kim et al.'s scheme [10] in Sections 2, 3, 4, 5, and 6, respectively. Finally, the concluding remarks are given in Section 7.

2 Tseng-Jan Group Signature Scheme I

2.1 Review of Tseng-Jan Scheme I

Tseng-Jan group signature scheme I [25] is based on discrete logarithm problem. We review this scheme in this subsection.

Setup. Let p and q be two large primes such that $q|(p-1)$, and g a generator with order q in $\mathbb{Z}_p$. Each group member U_i selects his secret key $x_i \in_R \mathbb{Z}_q^*$, and computes his public key $y_i := g^{x_i} \bmod p$. Similarly, the group manager (GM) selects his secret key $x \in_R \mathbb{Z}_q^*$, and computes his public key $y := g^x \bmod p$. Furthermore, GM selects a one-way hash function $h(\cdot)$. To join the group, a group member U_i sends his public key y_i to GM. Then, GM randomly chooses a random number $k_i \in_R \mathbb{Z}_q^*$, computes and sends back the following pair (r_i, s_i) to U_i privately:

$$r_i := g^{-k_i} \cdot y_i^{k_i} \bmod p, \quad s_i := k_i - r_i x \bmod q. \tag{1}$$

U_i can check the validity of his certificate (x_i, r_i, s_i) by

$$g^{s_i} y^{r_i} r_i \equiv (g^{s_i} y^{r_i})^{x_i} \bmod p. \tag{2}$$

Signing. To sign a message M, U_i first selects four random numbers $a, b, d, t \in_R \mathbb{Z}_q^*$, then calculates a signature (R, S, A, B, C, D, E) as follows:

$$\begin{aligned}
A &:= r_i^a \bmod p,\\
B &:= as_i - b \cdot h(A||C||D||E) \bmod q,\\
C &:= r_i a - d \bmod q,\\
D &:= g^b \bmod p,\\
E &:= y^d \bmod p,\\
\alpha_i &:= g^B y^C E D^{h(A||C||D||E)} \bmod p,\\
R &:= \alpha_i^t \bmod p,\\
S &:= t^{-1}(h(M||R) - Rx_i) \bmod q.
\end{aligned} \tag{3}$$

Verification. On receiving a signature (R, S, A, B, C, D, E) on a message M, a verifier first computes α_i as above and check the validity of the signature by

$$\alpha_i^{h(M||R)} \equiv (\alpha_i \cdot A)^R \cdot R^S \bmod p. \tag{4}$$

Note that the above equality holds since we have the following equations:

$$g^{s_i} y^{r_i} = g^{k_i} \bmod p, \quad \alpha_i = g^{ak_i} \bmod p, \quad \text{and} \quad \alpha_i A = \alpha_i^{x_i} \bmod p. \tag{5}$$

Open. To identify the signer of a valid group signature (R, S, A, B, C, D, E) on a message M, GM first computes the corresponding α_i and then find the signer by searching which pair (r_i, s_i, k_i) satisfies $\alpha_i \equiv (g^C \cdot E^{x^{-1}})^{r_i^{-1} \cdot k_i} \bmod p$, where x^{-1} and $r_i^{-1} \cdot k_i$ all are computed in $\mathbb{Z}_q$.

2.2 Security Analysis of Tseng-Jan Scheme I

Forging Signatures. Now we want to forge a group signature on an arbitrary message M even though we do not know any certificate, i.e., we need to find a tuple (R, S, A, B, C, D, E) that satisfies the following two verification equations:

$$\begin{cases} \alpha_i = g^B y^C E D^{h(A||C||D||E)} \bmod p, \\ \alpha_i^{h(M||R)} = (\alpha_i \cdot A)^R \cdot R^S \bmod p. \end{cases} \tag{6}$$

Note that in the generation of a signature, A, D, E and R all are some powers to the bases g and y. At the same time, C is embedded in the hash value $h(A||C||D||E)$. Therefore, we can define A, D, E, R as some known powers of g and y, and choose a value for C. Then, we try to solve B and S from equation (6). Hence, we choose nine numbers $a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, C \in \mathbb{Z}_q$ to define A, D, E and R as follows (all in $\mathbb{Z}_p$)

$$A := g^{a_1} y^{b_1}, \quad D := g^{a_2} y^{b_2}, \quad E := g^{a_3} y^{b_3}, \quad R := g^{a_4} y^{b_4}.$$

Then, we evaluate the two hash values $h := h(A||C||D||E)$, $h' := h(M||R)$, and replace the corresponding variables in equation (6) with the above expressions. Therefore, we get the following two equations for unknown variables of B and S:

$$\begin{cases} (B + a_3 + a_2 h)h' = (B + a_3 + a_2 h)R + a_1 R + a_4 S \bmod q, \\ (C + b_3 + b_2 h)h' \ = (C + b_3 + b_2 h)R + b_1 R + b_4 S \bmod q. \end{cases} \tag{7}$$

Therefore, if $b_4 \neq 0$ and $R \neq h' \bmod q$ (i.e., $R \neq h(M||R) \bmod q$.), we get the following solutions for S and B:

$$\begin{cases} S \ = b_4^{-1}[(C + b_3 + b_2 h)(h' - R) - b_1 R] \bmod q, \\ B = (a_1 R + a_4 S)(h' - R)^{-1} - (a_3 + a_2 h) \bmod q. \end{cases} \tag{8}$$

For summary, in the Tseng-Jan group signature scheme I [25], an attacker can forge a group signature on any message M as follows:

(1) Select nine random numbers $a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, C \in_R \mathbb{Z}_q$ such that $b_4 \neq 0$.
(2) Define $A := g^{a_1} y^{b_1}$, $D := g^{a_2} y^{b_2}$, $E := g^{a_3} y^{b_3}$, and $R := g^{a_4} y^{b_4}$ (all in $\mathbb{Z}_p$).
(3) Evaluate $h := h(A||C||D||E)$ and $h' := h(m||R)$.
(4) Determine if $R = h' \bmod q$. If yes, go to step (1); otherwise, continue.
(5) Compute S and B according to equation (8).
(6) Output (R, S, A, B, C, D, E) as a group signature on the message M.

The correctness of the above attack can be verified directly. When one such forged group signature is given, GM cannot find the signer. At the same time, note that in the above attack $R = h' \bmod q$ occurs only with a negligible probability since $h(\cdot)$ is a one-way hash function. Therefore, in general, our attack will succeed just by one try. Furthermore, for simplicity, some of those nine random numbers can be set as zeroes. For example, if we set $a_1 = b_2 = b_3 = a_4 = 0$, A, D, E and R can be computed simply: $A := y^{b_1} \bmod p$, $D := g^{a_2} \bmod p$, $E := g^{a_3} \bmod p$, $R := y^{b_4} \bmod p$. In such case, S and B can be computed by $S = b_4^{-1}(Ch' - CR - b_1 R) \bmod q$ and $B = -(a_3 + a_2 h) \bmod q$.

Forging Certificates. The authors of [11, 24] noted that for any group member U_i, (r_i, s_i) is a Nyberg-Rueppel signature [15] on message $y_i^{k_i}$. However, this *does not* imply that only GM can generate a valid certificate. Now, we demonstrate how to forge a certificate $(\bar{x}_i, \bar{r}_i, \bar{s}_i)$ that satisfies the equation (2). For this sake, we choose $a_0, b_0 \in \mathbb{Z}_q^*$, and define $\bar{r}_i := g^{a_0} y^{b_0} \bmod p$. Then, from equation (2), we have the following equation for unknown $\bar{x}_i$ and $\bar{s}_i$:

$$g^{\bar{s}_i} y^{\bar{r}_i} g^{a_0} y^{b_0} = (g^{\bar{s}_i} y^{\bar{r}_i})^{\bar{x}_i} \bmod p.$$

From the above equation, we get the following two equations for $\bar{x}_i$ and $\bar{s}_i$:

$$\bar{s}_i + a_0 = \bar{s}_i \cdot \bar{x}_i \bmod q, \quad \text{and} \quad \bar{r}_i + b_0 = \bar{r}_i \cdot \bar{x}_i \bmod q.$$

Therefore, we obtain the solutions for $\bar{x}_i$ and $\bar{s}_i$: $\bar{x}_i = 1 + b_0 \bar{r}_i^{-1} \bmod q$ and $\bar{s}_i = a_0 b_0^{-1} \bar{r}_i \bmod q$. The forged certificate $(\bar{x}_i, \bar{r}_i, \bar{s}_i)$ satisfies equation (2) since $g^{\bar{s}_i} y^{\bar{r}_i} \bar{r}_i = g^{a_0 b_0^{-1} \bar{r}_i} y^{\bar{r}_i} g^{a_0} y^{b_0} = g^{a_0 b_0^{-1} \bar{r}_i (1 + b_0 \bar{r}_i^{-1})} y^{\bar{r}_i (1 + b_0 \bar{r}_i^{-1})} = (g^{\bar{s}_i} y^{\bar{r}_i})^{\bar{x}_i} \bmod p$.

Now, an attacker can use the forged certificate $(\bar{x}_i, \bar{r}_i, \bar{s}_i)$ to generate valid group signature on any message M as a group member does. Firstly, the attacker chooses $a, b, d, t \in_R \mathbb{Z}_q^*$ and computes $A := \bar{r}_i^a \bmod p, B := a\bar{s}_i - b \cdot h(A||C||D||E) \bmod q, C := \bar{r}_i a - d \bmod q, D := g^b \bmod p$ and $E := y^d \bmod p$. Then, he computes $\bar{\alpha}_i := g^B y^C E D^{h(A||C||D||E)} = (\bar{\beta}_i)^a \bmod p$, where $\bar{\beta}_i := g^{\bar{s}_i} y^{\bar{r}_i} \bmod p$. Finally, he gets $R := \bar{\alpha}_i^t \bmod p$ and $S := t^{-1}[h(M||R) - R\bar{x}_i] \bmod q$. By using the facts that $\bar{\alpha}_i = (\bar{\beta}_i)^a \bmod p$ and $\bar{\alpha}_i A = (\bar{\beta}_i)^{a\bar{x}_i} \bmod p$, it is not difficult to verify that the resulting tuple (R, S, A, B, C, D, E) satisfies the verification equation (4), i.e., the forged group signature for the message M is valid.

Remark 1. The schemes proposed in [11, 24, 21] all are subject to similar attacks due to their similar structures. Especially, the above forged certificate can be directly used to generate valid group signatures in those schemes since all those schemes use the same certificate as in Tseng-Jan scheme I [25]. Compared with Joye's attacks [8] on the two schemes in [11, 24], our above attacks not only unify in a family, but also are very simple (especially for the forging certificate attack.). The attack on the Shi scheme [21], independently specified in [32] by Zhang et al, is a special case of our attacks. In addition, we notice that there is a design error in the Shi scheme. That is, all the following equations in [21] should be modified from modulo p to modulo q: eq. (5), eq. (6), eq. (11), eq. (15), and eq. (17). Otherwise, Shi scheme does not work since the signatueres generated by honest group members cannot be successfully validated by verifiers. Furthermore, with the above modification the Shi scheme seems as the same as the scheme by [24].

3 Tseng-Jan Group Signature Scheme II

3.1 Review of Tseng-Jan Scheme II

Tseng-Jan group signature scheme II [27] involves four parties: a trusted authority (TA), the group manager (GM), the group members, and the verifiers. TA acts as a third party to setup the system parameters. GM selects the group public/secret key pair. He (jointly with TA) issues certificates to new users who wants to join the group. Then, group members can anonymously sign on behalf of the group by using their membership certificates and verifiers check the validity of a group signature by using the group public key. In case of disputes, GM opens the contentious group signatures to reveal the identity of the actual signer.

System Initialization. In order to set up the system, TA sets a modulus $n = p_1 p_2$ where p_1 and p_2 are two large prime numbers (about 120 decimal digits) such that $p_1 = 3 \bmod 8$, $p_2 = 7 \bmod 8$, and $(p_1 - 1)/2$ and $(p_2 - 1)/2$ are smooth, odd and co-prime. Furthermore, $(p_1 - 1)/2$ and $(p_2 - 1)/2$ should contain several prime factors of about 20 decimal digits but no large prime factors. In this case, it is easy for TA to find the discrete logarithms for p_1 and p_2 [13, 14, 16, 17]. TA also defines e, d, v, t satisfying $ed = 1 \bmod \phi(n)$ and $vt = 1 \bmod \phi(n)$. Then, he selects an element g of large order in $\mathbb{Z}_n^*$, and computes $F := g^v \bmod n$. TA also chooses a hash function $h(\cdot)$. The public parameters of TA are $(n, e, g, F, h(\cdot))$, and the secret parameters of TA are (p_1, p_2, d, v, t). To create a group, GM selects a secret key x and computes the corresponding group public key $y := F^x \bmod n$.

When a user U_i (with identity information D_i) wants to join the group, TA and GM computes and sends the following s_i and x_i to U_i, respectively.

$$s_i := et \cdot \log_g ID_i \bmod \phi(n), \qquad \text{and} \qquad x_i := ID_i{}^x \bmod n. \tag{9}$$

where

$$ID_i := \begin{cases} D_i, & \text{if Jacobi symbol } (D_i|n) = 1; \\ 2D_i, & \text{if Jacobi symbol } (D_i|n) = -1. \end{cases} \tag{10}$$

The equation (10) guarantees the existence of the discrete logarithm of ID_i to the base g [14]. The membership certificate of the user U_i is (s_i, x_i).

Signing and Verification. To sign a message M, U_i first chooses two random integers r_1 and $r_2 \in \mathbb{Z}_n$. Then, U_i computes his group signature (A, B, C, D) on the message M as follows:

$$\begin{aligned} A &:= y^{r_1} \bmod n \\ B &:= y^{r_2 e} \bmod n \\ C &:= s_i + r_1 \cdot h(M||A||B) + r_2 e \\ D &:= x_i \cdot y^{r_2 \cdot h(M||A||B||C)} \bmod n. \end{aligned} \tag{11}$$

Upon receiving a signature tuple (A, B, C, D) on message M, a verifier can verify the validity of this signature by checking whether

$$D^e A^{h(M||A||B)} B \equiv y^C B^{h(M||A||B||C)} \bmod n. \tag{12}$$

Open. GM with the secret key x can identify the signer of a signature by finding the ID_i that satisfies the following equation:

$$(ID_i)^{xe} \equiv D^e \cdot B^{-h(M||A||B||C)} \bmod n. \tag{13}$$

3.2 Security Analysis of Tseng-Jan Scheme II

In [27], Tseng and Jan provide detailed security analysis to demonstrate that their scheme is secure against forgeries and that the anonymity of the signer in their scheme depends on computing the discrete logarithm modulo for the composite number n. However, our analysis in this subsection shows that Tseng-Jan scheme [27] is linkable and universally forgeable.

Linkability. It is easy to see that the value in the left side of equation (13) is an invariant for user U_i since ID_i is the identity information derived from user U_i's real identity, and x, e both are public information. Therefore, given two valid group signatures, (A, B, C, D) and $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$, on messages M and $\bar{M}$, respectively, anybody (not necessarily group member) can determine whether they are signed by the same group member by checking whether the following equality holds:

$$D^e B^{-h(M||A||B||C)} \equiv \bar{D}^e \bar{B}^{-h(\bar{M}||\bar{A}||\bar{B}||\bar{C})} \bmod n.$$

The above equality shows that Tseng-Jan scheme [27] is linkable. Similarly, the scheme in [27] is also linkable.

Forging Signatures. Note that in [26], the value D in equation (11) is computed in a different way: $D := x_i \cdot y^{r_2 \cdot h(M||A||B)} \bmod n$. However, this modification does not improve the security of Tseng-Jan scheme II. Similar to what we did in Section 2.2, we want to forge a group signature for an arbitrary message M even without any membership certificate. Note that the verification equation (12) is about some powers of A, B, D and y. So we first define A, B, D as some known powers to the base y, and then try to solve C from equation (12). Therefore, we choose three random number r_1, r_2, r_4 and define A, B, D as follows (A and B have the same forms as in equation (11)):

$$A := y^{r_1} \bmod n; \quad B := y^{r_2 e} \bmod n; \quad D := y^{r_4} \bmod n.$$

Then, from the verification equation (12), we get the condition for the value C:

$$r_4 e + r_1 \cdot h(M||A||B) + r_2 e = C + r_2 e \cdot h(M||A||B||C) \bmod \phi(n). \tag{14}$$

We have selected r_1, r_2 and r_4, so A, B, D and then hash value $h(M||A||B)$ all are fixed. Therefore, finding a solution for unknown value C from equation (14) seems difficult because we do not know the modulus $\phi(n)$ and the value of C is embedded in the hash value $h(M||A||B||C)$. However, we note that solving equation (14) seems really difficult only if r_1, r_2 and r_4 are truly selected as *random* numbers. But, we are attackers. So we have the freedom to choose some special values for r_1, r_2 and r_4. In other words, to get a solution for the value C, we can let those numbers satisfy some specific relationships. it is not difficult to find the following solution for equation (14):

$$C := r_1 \cdot h(M||A||B) + r_2 e \in \mathbb{Z}^+; \quad r_4 := r_2 \cdot h(M||A||B||C) \in \mathbb{Z}^+.$$

Now, we summary our attack on the Tseng-Jan scheme II [27] as follows:

(1) Firstly, select two random numbers r_1, r_2.
(2) Then define $A := y^{r_1} \bmod n$, and $B := y^{r_2 e} \bmod n$.

(3) Compute $C := r_1 \cdot h(M||A||B) + r_2 e \in \mathbb{Z}^+$.
(4) Define $r_4 := r_2 \cdot h(M||A||B||C) \in \mathbb{Z}^+$, and then compute $D := y^{r_4} \bmod n$.
(5) Output (A, B, C, D) as a group signature on the message M.

It is easy to check that the above attack is correct. At the same time, when such a forged signature is given, the group manager cannot find any group member to take responsible for it.

In fact, if we choose a new random number r_3, the values of C and D in the above attack can be randomized by defining C and r_4 as follows

$$C := r_1 \cdot h(M||A||B) + r_2 e + r_3 e \in \mathbb{Z}^+; \quad r_4 := r_2 \cdot h(M||A||B||C) + r_3 \in \mathbb{Z}^+.$$

Furthermore, we have another idea to solve equation (14): First define A, B and C, then calculate hash values of $h(M||A||B)$ and $h(M||A||B||C)$, and finally solve r_4 for D. However, it seems difficult to find the value of r_4 from equation (14) since we do not know the values of modulus $\phi(n)$ and $e^{-1} \bmod \phi(n)$. But we notice that we can find a value for r_4 if e can be eliminated from equation (14). Here is the trick. We use $r_1 e$ to replace r_1 (i.e., $A := y^{r_1 e} \bmod p$) and define $C := r_3 e$ (in $\mathbb{Z}$) for some random number r_3, then r_4 can be attained:

$$r_4 := r_3 + r_2 \cdot h(M||A||B||C) - r_1 \cdot h(M||A||B) - r_2 \in \mathbb{Z}.$$

Forging Certificates. Note that the membership certificates in [26] and [27] are the same. Therefore, according to equation (9), for any positive random integer k there are two ways to forge valid membership certificates: (1) A group member U_i can generate a new certificate $(ks_i, x_i^k \bmod n)$ using his certificate (s_i, x_i); (2) Anybody (not necessarily a group member) can use $(\bar{s}_i = ke, \bar{x}_i = y^k \bmod n)$ as a valid certificate [9]. Given a valid group signature generated by using such forged certificates, GM cannot identify the signer.

Later, Popescu proposed a modification of the Tseng-Jan scheme II in [18]. However, in the next section, we will show that Popescu's scheme is still insecure.

4 Popescu's Group Signature Scheme

4.1 Review of Popescu's Scheme

Key Generation. The TA selects two large primes p_1, p_2 as in [27] (see §3.1) and sets $n := p_1 p_2$. Then, the TA selects g of large order in $\mathbb{Z}_n^*$, a large integer e (160 bits) such that $\gcd(e, \phi(n)) = 1$, and then computes d satisfying $de = 1 \bmod \phi(n)$. The GM chooses a secret key x and computes the corresponding public key $y := g^x \bmod n$. The GM also chooses a collision-resistant hash function $h(\cdot)$. The public parameters are (n, e, g, y, h), the TA's secret key is (p_1, p_2, d) and GM's secret key is x.

When a user U_i with identity information $ID_i \in \mathbb{Z}_n$ wants to join the group, the TA and GM compute the following s_i and x_i, respectively

$$s_i := {ID_i}^d \bmod n, \quad x_i := (ID_i + eg)^x \bmod n.$$

Then, the membership certificate (s_i, x_i) is sent to the user U_i securely.

Signing. To sign a message M, the user U_i chooses two random numbers r_1, r_2, and then computes his group signature (A, B, C, D) as follows

$$\begin{aligned} A &:= y^{r_2 e} \bmod n \\ B &:= x_i y^{s_i + r_1} \bmod n \\ C &:= x_i y^{r_2} \bmod n \\ D &:= s_i h(M||A) + r_1 h(M||A). \end{aligned} \tag{15}$$

Verification. (A, B, C, D) is a valid group signature on message M iff the following equality holds:

$$C^{eh(M||A)} y^{eD} \equiv B^{eh(M||A)} A^{h(M||A)} \bmod n. \tag{16}$$

Open. Finally, the GM can recover the signer of a signature (A, B, C, D) on message M by checking which identity ID_i satisfies

$$(ID_i + eg)^{xe} \equiv C^e A^{-1} \bmod n. \tag{17}$$

4.2 Security Analysis of Popescu's Scheme

In [18], Popescu claimed that his scheme is unforgeable and unlinkable since a non-group member (including the TA and the GM) does not have a valid membership certificate (s_i, x_i) and deciding the linkability of two group signatures is computationally hard under decisional Diffie-Hellman assumption.

However, these claims are not true. In this subsection, we will show that in Popescu's scheme, (1) Deciding the linkability of two group signatures and forging a valid group signature on any message are easy even for a non-group member; (2) Any two random numbers can be used as a valid membership certificate; and (3) GM can forge valid group signatures on behalf of any group member. In other words, Popescu's scheme is *linkable, universal forgery* and does not satisfy *traceablility, coalition-resistance* and *exculpability.*

Linkability. First of all, it is easy to see that the left side of equation (17) is an invariant for user U_i. Therefore, given two valid group signatures (A, B, C, D) and $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$, by checking the following equality, anybody can determine whether they are signed by the same group member:

$$C^e A^{-1} \equiv \bar{C}^e \bar{A}^{-1} \bmod n.$$

Forging Signatures. Now, we want to forge a group signature on an arbitrary given message by using similar method as we used in previous sections, even if we do not know any member certificate (s_i, x_i). Since the verification equation (16) is about some powers of A, B, C and y, we choose three random numbers r_1, r_2, r_3 and define A, B, C as follows (A has the same form as in equation (15)):

$$A := y^{r_2 e} \bmod n, \quad B := y^{r_1} \bmod n, \quad C := y^{r_3} \bmod n.$$

Let $h = h(M||A)$. Then, from the verification equation (16), we get the condition for the value D: $r_3 eh + De = r_1 eh + r_2 eh \bmod \phi(n)$, i.e:

$$r_3 h + D = r_1 h + r_2 h \bmod \phi(n). \tag{18}$$

Of course, we do not know the modulus $\phi(n)$, but equation (18) has a trivial solution $D := (r_1 + r_2 - r_3)h \in \mathbb{Z}^+$ if we choose r_1, r_2, r_3 such that $r_1 + r_2 > r_3$. This shows that Popsecu's scheme is universally forgeable. In summary, an attacker can forge a Popescu's ID-based group signature [18] on any message M as follows:

(1) First select three random numbers r_1, r_2, r_3 such that $r_1 + r_2 > r_3$.
(2) Then define $A := y^{r_2 e} \bmod n$, $B := y^{r_1} \bmod n$, and $C := y^{r_3} \bmod n$.
(3) Compute $h := h(M||A)$, and $D := (r_1 + r_2 - r_3)h \in \mathbb{Z}^+$.
(4) Output (A, B, C, D) as a valid group signature on message M.

Forging Certificates. We now want to derive the determining equation for a valid membership certificate. Let $(\bar{s}_i, \bar{x}_i)$ be a pair of two random numbers. We select two random numbers r_1, r_2 and compute (A, B, C, D) according to equation (15), as if we have a valid member certificate. Let $h = h(M||A)$. Then, we calculate the both sides of the verification equation as follows:

$$C^{eh} y^{eD} = (\bar{x}_i y^{r_2})^{eh} \cdot y^{eh(\bar{s}_i + r_1)} = (\bar{x}_i)^{eh} \cdot y^{(\bar{s}_i + r_1 + r_2)eh} \bmod n,$$
$$B^{eh} A^h = (\bar{x}_i y^{\bar{s}_i + r_1})^{eh} \cdot (y^{r_2 e})^h = (\bar{x}_i)^{eh} \cdot y^{(\bar{s}_i + r_1 + r_2)eh} \bmod n.$$

Obviously, they are identical. Therefore, we reveal an unbelievable fact: In Popsecu's modified scheme [18], any random number pair $(\bar{s}_i, \bar{x}_i)$ is a valid membership certificate!

No Exculpablility. Above fact not only strengthens the conclusion that Popsecu's scheme is universally forgeable, but also reveals another fact that Popsecu's scheme has no exculpablility: The group manager, who knows the secret value x_i for user U_i, can generate a valid group signature for any message on behalf of U_i by using $(x_i, \bar{s}_i)$ as a membership certificate, where $\bar{s}_i$ is chosen as a random number. If such a valid group signature (A, B, C, D) is opened, user U_i will be identified because $x_i^e = (ID_i + eg)^{xe} = C^e A^{-1} \bmod n$.

5 Xia-You Group Signature Scheme

5.1 Review of Xia-You Scheme

Setup of Trusted Authority (TA). TA generates two prime numbers p_1 and p_2 satisfying the same conditions listed in the Setup of Tseng-Jan scheme II and sets $m := p_1 p_2$. In this case, it is easy for TA to find the discrete logarithms modulo p_1 and p_2. An integer g is chosen such that $g < \min\{p_1, p_2\}$. Finally, TA publishes (m, g) but keeps the prime factors p_1 and p_2 as his secret.

Generating Private Keys. Since a signer U_i's identity information D_i (which is smaller than m) is not guaranteed to have a discrete logarithm modulo the composite number m, TA computes ID_i by equation (10) (respect to modulus m). Now TA computes the private key x_i for U_i as the discrete logarithm of ID_i to the base g:

$$ID_i = g^{x_i} \bmod m. \tag{19}$$

Finally, TA sends x_i to U_i in a secure way and U_i can check the validity of x_i by verifying equation (19). The reader can refer to [13, 14] for details.

Setup of Group Manager (GM). GM chooses two large primes p_3 and p_4 such that $p_3 - 1$ and $p_4 - 1$ are not smooth, and sets $n = p_3 p_4$ such that $n > m$. Let e be an integer satisfying $\gcd(e, \phi(n)) = 1$, and computes d such that $ed = 1 \bmod \phi(n)$. Then, GM chooses two integers $x \in \mathbb{Z}_m, h \in \mathbb{Z}_m^*$, and then computes $y := h^x \bmod m$ as the group public key. Let $H(\cdot)$ be a collision-resistant hash function that maps $\{0,1\}^*$ to $\mathbb{Z}_m$. The group public key is (n, e, h, y, H) and GM's secret key is (x, d, p_3, p_4).

Generating Membership Keys. When a signer U_i wants to join the group, GM computes the membership key z_i of U_i as follows

$$z_i = ID_i^d \bmod n. \tag{20}$$

Then, z_i is sent to U_i in a secure way and U_i checks the validity of z_i by verifying $ID_i = z_i^e \bmod n$.

Signing. To sign a message M, U_i first chooses five random numbers $\alpha, \beta, \theta, \omega \in \mathbb{Z}_m$ and $\delta \in \mathbb{Z}_n$, and then computes the signature (A, B, C, D, E, F, G) as follows:

$$\begin{aligned}
&A : = y^{\alpha} \cdot z_i \bmod n,\\
&B : = y^{\omega} \cdot ID_i,\\
&C : = h^{\omega} \bmod m,\\
&D : = H(y||g||h||A||B||\hat{B}||C||v||t_1||t_2||t_3||M),\\
&E : = \delta - D(\alpha e - \omega),\\
&F : = \beta - D\omega,\\
&G : = \theta - Dx_i,
\end{aligned} \tag{21}$$

where

$$\begin{aligned}
&\hat{B} := B \bmod m, \quad v := (A^e/B) \bmod n;\\
&t_1 := y^{\delta} \bmod n, \quad t_2 := y^{\beta} \cdot g^{\theta} \bmod m, \quad t_3 := h^{\beta} \bmod m.
\end{aligned}$$

Verification. A verifier accepts a signature (A, B, C, D, E, F, G) on a message M if and only if

$$D \equiv H(y||g||h||A||B||\hat{B}||C||v||t_1'||t_2'||t_3'||M), \tag{22}$$

where $\hat{B}$ and v are computed as in signing equation, i.e., $\hat{B} = B \bmod m, v = (A^e/B) \bmod n$, but t_1', t_2' and t_3' are given by the following equations

$$t_1' := v^D y^E \bmod n, \quad t_2' := \hat{B}^D y^F g^G \bmod m, \quad t_3' := C^D h^F \bmod m. \tag{23}$$

Open. Given a valid group signature (A, B, C, D, E, F, G) on a message M, the group manager can identify the signer by finding the ID_i such that

$$ID_i = B \cdot C^{-x} \bmod m.$$

5.2 Security Analysis of Xia-You Scheme

Xia and You claimed that their scheme [29] satisfies all the security properties listed in Section 1. However, in this subsection presents several attacks to show that Xia-You scheme [29] is insecure.

Linkability. From signing equation (21), we know that E, F, G are three integers (may be negative), and B is a non-negative integer. Since $B = y^w \cdot ID_i$, we know

$ID_i|B$ if U_i is the signer of a valid group signature (A, B, C, D, E, F, G). Therefore, for anyone who knows the identities of group members, he can find the signer with a high probability. Usually, ID_i, a large integer (e.g. 160 bits), is computed as a hash value of U_i's real-world identity (e.g., names, network address, etc.). So it seems unlikely that there are two ID_i and ID_j such that $ID_i|B \in \mathbb{Z}$ and $ID_j|B \in \mathbb{Z}$. Hence, we can say that Xia-You scheme only satisfy *weak* anonymity and unlinability. Because, for anyone (especially group members) who know the identities of all group members, Xia-You scheme does not satisfy the anonymity and unlinability. In addition, in the situation without knowledge of the identities of group members, it is also possible to break the linkability by using the great common divisor of several values of B's.

Forging Signatures. Using similar method used in the previous sections, we can forge a group signature on an arbitrary given message M even without any membership certificate (ID_i, x_i, z_i). Note that to satisfy the verification equations (22) and (23), we can first choose A, B, C and t_1, t_2, t_3, then we get D by evaluating the corresponding hash value, and finally try to solve the values of E, F and G from equation (23). If we observe equations (21)-(23) carefully, we will know that a good strategy is to choose A, B, t_1 and t_2 as some known representations of bases y and g, but C and t_3 as powers of h. Therefore, we can choose ten random numbers $a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5$ to define A, B, C and t_1, t_2, t_3 as follows:

$$\begin{array}{lcl} A := y^{a_1} \cdot g^{a_2} \bmod n, & & t_1 := y^{b_1} \cdot g^{b_5} \bmod n, \\ B := y^{a_3} \cdot g^{a_4}, & \text{and} & t_2 := y^{b_2} \cdot g^{b_3} \bmod m, \\ C := h^{a_5} \bmod m. & & t_3 := h^{b_4} \bmod m. \end{array}$$

Then, we compute $\hat{B} := B \bmod m$, $v := (A^e/B) \bmod n = y^{a_1 e - a_3} \cdot g^{a_2 e - a_4} \bmod n$ and evaluate the hash value $D = H(y||g||h||A||B||\hat{B}||C||v||t_1||t_2||t_3||M)$. At last, to get the values of E, F and G, we replace the occurrences of $t_1', t_2', t_3', \hat{B}$ and v in equations (23) by t_1, t_2, t_3, B and $y^{a_1 e - a_3} \cdot g^{a_2 e - a_4} \bmod n$, respectively, and then we have

$$\begin{array}{l} b_1 = (a_1 e - a_3) D + E \bmod \phi(n), \\ b_5 = (a_2 e - a_4) D \bmod \phi(n), \\ b_2 = a_3 D + F \bmod \phi(m), \\ b_3 = a_4 D + G \bmod \phi(m), \\ b_4 = a_5 D + F \bmod \phi(m). \end{array}$$

In general, we cannot find a solution for (E, F, G) from the above equation system. However, we can set the ten numbers, i.e., $a_1, \cdots, b_5$, satisfying specific relationships such that the above equation system has one solution. First, please note that we should set $b_5 = 0$. Because D is determined by those ten numbers, we cannot require $b_5 = (a_2 e - a_4) D \bmod \phi(n)$ again. $b_5 = 0$ also implies that $a_2 e - a_4 = 0$, i.e., $a_4 = a_2 e$ (in $\mathbb{Z}$). Secondly, we notice that F has to satisfy the third and the fifth equations at the same time, so we should set these two equations as the same one. This means that $a_5 = a_3$ and $b_4 = b_2$. Therefore, under the conditions of $b_5 = 0$, $a_4 = a_2 e$, $a_5 = a_3$ and $b_4 = b_2$, we get the following solution for (E, F, G) even though we do not know the values of $\phi(m)$ and $\phi(n)$:

$$E := b_1 + (a_3 - a_1 e) D \in \mathbb{Z}, \quad F := b_2 - a_3 D \in \mathbb{Z}, \quad G := b_3 - a_2 e D \in \mathbb{Z}.$$

In summary, an attacker can forge a valid group signature on a message M as follows:

(1) First of all, select six random numbers $a_1, a_2, a_3, b_1, b_2, b_3$.
(2) Then, define $A := y^{a_1} \cdot g^{a_2} \bmod n$, $B := y^{a_3} \cdot g^{a_2 e}, C := h^{a_3} \bmod m, t_1 := y^{b_1} \bmod n, t_2 := y^{b_2} \cdot g^{b_3} \bmod m, t_3 := h^{b_2} \bmod m$.
(3) Compute $\hat{B} := B \bmod m$ and $v := (A^e/B) \bmod n = y^{a_1 e - a_3} \bmod n$, and then $D := H(y||g||h||A||B||\hat{B}||C||v||t_1||t_2||t_3||M)$.
(3) Compute $E := b_1 + (a_3 - a_1 e)D \in \mathbb{Z}, F := b_2 - a_3 D \in \mathbb{Z}$, and $G := b_3 - a_2 e D \in \mathbb{Z}$.
(3) Output (A, B, C, D, E, F, G) as a group signature on the message M.

Again, it is not difficult to verify that the above attack is successful.

Forging Certificates. Similarly, we can get the following conditions for a valid membership certificate $(\overline{ID}_i, \bar{x}_i, \bar{z}_i)$:

$$\bar{z}_i^e = \overline{ID}_i \bmod n, \quad \text{and} \quad \overline{ID}_i = g^{\bar{x}_i} \bmod m.$$

These two conditions are the exact equations (19) and (20). So, it seems that valid membership certificates can only be generated jointly by TA and GM. However, for any non-negative integer k, it is not difficult to see that (1) A group member U_i with membership certificate (ID_i, x_i, z_i) can generate a valid membership certificates $(ID_i^k, kx_i, z_i^k \bmod n)$, and (2) anyone (not necessarily a group member) can use $(\overline{ID} := g^{ke}, \bar{x} := ke, \bar{z} := g^k \bmod n)$ as a valid certificate to generate group signature on any message. Given a valid signature generated by using such forged membership certificate, of course, GM cannot identify the signer.

Remarks 2. Different attacks on Xia-You scheme are also identified independently by Zhang et al. in [30], and Zhang et al. [31]. The attack in [30] is a special case of our forging signatures, and the two attacks in [31] are weaker than our (universally) forging certificate attack since their attacks can only be mounted by colluding group members.

6 Kim-Park-Won Convertible Group Signature Scheme

6.1 Review of Kim-Park-Won Scheme

To set up a system, the GM first chooses three primes p', q', f such that $p := 2fp' + 1$ and $q := 2fq' + 1$ are also primes. Then, the GM sets $n = p\,q$ and selects an element $g \in \mathbb{Z}_n^*$ of order f, i.e., $g^f = 1 \bmod n$. Furthermore, the GM chooses $\gamma \in \mathbb{Z}_{\phi(n)}^*$ and computes d such that $\gamma d = 1 \bmod \phi(n)$. Let ID_G be the identity information of the group, $h(\cdot)$ a secure hash function. Finally, the GM makes $(n, \gamma, f, g, h(\cdot), ID_G)$ as public information, (d, p', q') as his private key.

To joint the group, a user U_i with identity information ID_i chooses a random secret number $s_i \in (0, f)$, then computes $y_i := g^{s_i}$ and sends (ID_i, y_i) to the GM. Then, the GM computes and sends following x_i to U_i securely:

$$x_i := (ID_G \cdot y_i)^{-d} \bmod n. \tag{24}$$

At the same time, to identify signers in case of disputes, the GM stores (ID_i, y_i, x_i) into a complete list for all registered group members.

To generate a group signature (e, z_1, z_2) on message M, user U_i first chooses two random numbers $r_1 \in_R [0, f), r_2 \in_R [0, n)$ and then computes:

$$\begin{aligned} V &: = g^{r_1} r_2^{\gamma} \mod n \\ e &: = h(V||M) \\ z_1 &: = r_1 + s_i e \mod f \\ z_2 &: = r_2 {x_i}^{e} \mod n. \end{aligned} \tag{25}$$

To verify a group signature (e, z_1, z_2), a verifier checks whether

$$e \equiv h(\bar{V}||M), \quad \text{where } \bar{V} := (ID_G)^e g^{z_1} {z_2}^{\gamma} \bmod n. \tag{26}$$

To open a valid group signature (e, z_1, z_2) for message M, the GM first calculates $\bar{V} := (ID_G)^e g^{z_1} {z_2}^{\gamma} \bmod n$, and then searches his list of all (ID_j, y_j, x_j) to find the signer U_i if U_i's (x_i, y_i) satisfies the following equality

$$g^{z_1} \equiv \bar{V} \cdot z_2^{-\gamma} \cdot x_i^{e\gamma} \cdot y_i^e \bmod n. \tag{27}$$

6.2 Security Analysis of Kim-Park-Won Scheme

Forging Signatures. Now, we first try to forge a valid group signature on any given message M under the assumption that we do not know any valid membership certificate. Note that the verification equation is to evaluate a hash value, and we have assumed that $h(\cdot)$ is a secure hash function. Therefore, if we first choose value for e, it seems difficult to find a tuple (V, z_1, z_2) such that both relations in verification equation (26) are satisfied. So we go in the other direction, i.e., we first choose a value for V and calculate $e := h(V||M)$, then we try to find a pair (z_1, z_2) satisfying the following equality:

$$V \equiv (ID_G)^e g^{z_1} {z_2}^{\gamma} \bmod n.$$

Note that the above equation is about several powers of ID_G, g and z_2, so we choose four numbers, a_1, a_2, b_1, b_2, and then define V and z_2 as follows

$$V := (ID_G)^{a_1} g^{b_1} \mod n, \quad z_2 := (ID_G)^{a_2} g^{b_2} \mod n.$$

Replacing all occurrences of V and z_2 in equation (26) with the above two expressions, respectively, we get the following equation:

$$(ID_G)^{a_1} g^{b_1} \equiv (ID_G)^{e + a_2\gamma} g^{z_1 + b_2\gamma} \bmod n.$$

Then, we have

$$\begin{cases} a_1 = e + a_2\gamma \bmod ord(ID_G) \\ b_1 = z_1 + b_2\gamma \bmod f \end{cases}, \quad \text{or} \quad \begin{cases} a_1 = e + a_2\gamma \bmod \phi(n) \\ b_1 = z_1 + b_2\gamma \bmod f \end{cases}. \tag{28}$$

Where $ord(ID_G)$ denotes the multiplicative order of element $ID_G \in \mathbb{Z}_n^*$, and $e := h(V||M) = h(ID_G^{a_1} g^{b_1} \bmod n||M)$.

In the above two equation systems, given a_1, b_1 (and then V, e), finding solutions for b_2 and z_1 are very easy since modulus f is known. However, finding a solution for

a_2 seems difficult since we do not know any value of $ord(ID_G), \phi(n)$, γ^{-1} mod $\phi(n)$ or γ^{-1} mod $ord(ID_G)$. But, in the following three special settings, we can find some solutions.

(1) $ID_G^{2f} = 1 \bmod n$, i.e., $ord(ID_G) = 2, f$, or $2f$. In this case, an attacker can forge valid group signature by setting $a_2 := (a_1 - e)\gamma^{-1} \bmod ord(ID_G)$. This is the attack pointed out in [19]. However, if the suggested parameters are used, i.e., $|p'| = |q'| \approx 234$ and $|f| \approx 160$ [10], we note that this case occurs only with a negligible probability $(4f^2 - 1)/n < 1/2^{466}$.

(2) Since the GM knows the value of $\phi(n)$, he can generate a valid group signature by setting $a_2 := (a_1 - e)\gamma^{-1} \bmod \phi(n)$. In fact, this is a trivial result. Because in general group signature schemes, including Kim-Park-Won scheme, GM always can create nonexistent membership certificate and generate group signature.

(3) $ID_G^d \bmod n$ is known. In this case, if we define $z_2 := (ID_G^d)^{\bar{a}_2} g^{b_2} \bmod n$, then the equation for $\bar{a}_2$ will become:

$$a_1 = e + \bar{a}_2 \cdot d\gamma \bmod \phi(n).$$

Since $d\gamma = 1 \bmod \phi(n)$, one trivial solution is attained $\bar{a}_2 := a_1 - e \in \mathbb{Z}^+$ if $a_1 - e > 0$. If we assume $h(\cdot) \leq l$ and choose a_1 such that $a_1 \geq 2^l$, we will always have $a_1 - e > 0$. However, how to get the value of $ID_G^d \bmod n$? The methods are given in the next part.

Forging Certificates. A valid membership certificate is defined by equation (24), which is a RSA signature of GM on the message $(ID_G \cdot y_i)^{-1}$. However, this does not imply that valid membership certificates can only be generated by the GM. It is easy to know that the following equation defines a valid membership certificate $(\bar{s}_i, \bar{x}_i)$ too, since it is a variant of equation (24):

$$ID_G \cdot g^{\bar{s}_i} \cdot \bar{x}_i^{\gamma} = 1 \bmod n. \tag{29}$$

Let U_i and U_j, with certificates (s_i, x_i) and (s_j, x_j) respectively, be two colluding group members, then they have several ways to forge a valid membership certificate $(\bar{s}, \bar{x})$.

(a) For any integer $k > 1$, define $\bar{s} := ks_i - (k-1)s_j \bmod f$ and $\bar{x} := x_i^k \cdot x_j^{-(k-1)} \bmod n$. This method works since $\bar{x} := x_i^k \cdot x_j^{-(k-1)} = (ID_G \cdot g^{ks_i - (k-1)s_j})^{-d} = (ID_G \cdot g^{\bar{s}})^{-d} \bmod n$.

(b) If they choose an integer $\delta > 0$ and define $s_j := s_i + \delta \bmod f$, they can get the value of $g^{\delta d}$ by $g^{\delta d} := x_i \cdot x_j^{-1} \bmod n$. Then, for any integer $k > 1$, define $\bar{s} := s_i + k\delta \bmod f$ and $\bar{x} := x_i \cdot (g^{\delta d})^{-k} \bmod n$. $(\bar{s}, \bar{x})$ is a valid certificate because $\bar{x} = x_i \cdot (g^{\delta d})^{-k} = (ID_G \cdot g^{s_i})^{-d}(g^{\delta k})^{-d} = (ID_G \cdot g^{s_i + k\delta})^{-d} = (ID_G \cdot g^{\bar{s}})^{-d} \bmod n$. Specifically, if $\delta = 1$, then we get $g^d = x_i \cdot x_j^{-1} \bmod n$ and $(ID_G)^{-d} = x_i \cdot (g^d)^{s_i} \bmod n$; if $\delta = s_i$, i.e., $s_j = 2s_i \bmod f$, we get $(ID_G)^{-d} = (x_i)^2 \cdot x_j^{-1} \bmod n$. Therefore, $ID_G^d \bmod n$ is available.

(c) If they set $s_i := ab$ and $s_j := ab + b$ for two known positive integers a and b, g^{bd} can be attained by computing $x_i \cdot x_j^{-1} \bmod n$, and then ID_G^{-d} can be attained by computing $x_i \cdot (g^{bd})^a \bmod n$. When g^{bd} and ID_G^{-d} are known, they can generate a valid certificate $(\bar{s}, \bar{x})$ by defining $\bar{s} := bk \bmod f$ and $\bar{x} := ID_G^{-d} \cdot (g^{bd})^{-k} \bmod n$, for any integer $k > 1$. This attack was first found by Lim and Lee [12].

In the above three cases, two colluding group members are needed. However, if the system allows a user own two certificates at the same time or an old group member can get a new certificate when he joins the same system for the second time, a group member alone can mount above attacks successfully.

Signer Identification. For a valid group signature (e, z_1, z_2) on message M, if replacing the occurrence of $\bar{V}$ in equation (27) by $ID_G^e g^{z_1} z_2^\gamma \bmod n$, we have

$$g^{z_1} \equiv ID_G^e \cdot g^{z_1} \cdot z_2^\gamma \cdot z_2^{-\gamma} \cdot x_i^{e\gamma} \cdot g^{s_i e} \bmod n,$$

i.e., $1 = (ID_G \cdot g^{s_i} \cdot x_i^\gamma)^e \bmod n$. However, according to equation (29), we know $1 \equiv ID_G \cdot g^{s_i} \cdot x_i^\gamma$ for every certificate (s_i, x_i). This shows that given a valid group signature, equation (27) is an equality for all certificates (s_i, x_i). In other words, equation (27) cannot be used to identify the signer because all certificates (s_i, x_i) satisfy it. Wang et. al first pointed out this problem [28], but they have no explanations for it. Now we point out the reason: If (e, z_1, z_2) is U_i's valid group signature on message M, it is also U_j's valid group signature on message M. More specifically, we denote $\delta := s_j - s_i \bmod f$ and assume that U_i chooses two random numbers r_1 and r_2 to generate his signature (e, z_1, z_2) as in equation (25). Then, it is easy to check that (e, z_1, z_2) is also a valid signature of U_j for the same message if U_j chooses $\bar{r}_1 := r_1 - \delta e \bmod f$ and $\bar{r}_2 := r_2 g^{\delta d e} \bmod n$ as his own two random numbers and then generates his signature. Therefore, for the same message, the signature spaces of any two group member are the same. So, it is impossible (in information theoretic sense) to trace the signer even for the GM. Therefore, Kim-Park-Won scheme [10] is totally anonymous and unlinkable even for the GM.

7 Concluding Remarks

In this paper, by using similar methods, we successfully identified different universally forging attacks on several group signature schemes proposed in [25, 27, 18, 29, 10]. Using our attacks, anybody (not necessarily a group member) can forge valid group signature on any message. Therefore, all these group signature schemes are insecure. At the same time, our attacks also implied that constructing group signatures by the ad-hoc methods should be terminated. Our attacking method described in this paper might be used as a security test in the future design of group signatures (and other possible signature schemes).

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In: *Crypto'2000, LNCS 1880*, pp. 255-270. Springer-Verlag, 2000.
2. G. Ateniese and G. Tsudik. Some open issues and new directions in group signature schemes. In: *Financial Cryptography* (*FC'99*), *LNCS 1648*, pp. 196-211. Springer-Verlag, 1999.
3. S. Brands. An efficient off-line electronic cash systems based on the representation problem. *Technical Report CS-R9323*, Centrum voor Wiskunde en Inforamatica, April 1993.

4. J. Camenisch and M. Stadler. Effient group signature schemes for large groups. In: *Crypto'97, LNCS 1294*, pp. 410-424. Springer-Verlag, 1997.
5. J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In: *Asiacrypt'98, LNCS 1514*, pp. 160-174. Springer-Verlag, 1998.
6. C.-C. Chang, and K.-F. Hwang. Towards the forgery of a group signature without knowing the group center's secret. In: *Information and Communications Security* (*ICICS'01*), *LNCS 2229*, pp. 47-51. Springer-Verlag, 2001.
7. D. Chaum and E. van Heyst. Group Signatures. In: *Eurocrypt'91, LNCS 950*, pp. 257-265. Springer-Verlag, 1992.
8. M. Joye, N.-Y. Lee, and T. Hwang. On the security of the Lee-Chang group signature scheme and its derivatives. In: *Information Security* (*ISW'99*), *LNCS 1729*, pp. 47-51. Springer-Verlag 1999.
9. M. Joye, S. Kim, and N.-L. Lee. Cryptanalysis of two group signature schemes. In: *Information Security* (*ISW'99*), *LNCS 1729*, pp. 271-275. Springer-Verlag 1999.
10. S.J. Kim, S.J. Park, and D.H Won. Convertible group signatures.In: *Asiacrypt'96, LNCS 1163*, pp. 311-321. Springer-Verlag, 1996.
11. W. Lee and C. Chang. Efficient group signature scheme based on the discrete logarithm. *IEE Proc. Comput. Digital Techniques*, 1998, 145 (1): 15-18.
12. C.H. Lim and P.J. Lee. Remarks on convertible signatures of Asiacrypt'96. *Electronics Letters*, 1997, 33(5): 383-384.
13. U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In: *Eurocrypt'91, LNCS 547*, pp. 498-507. Springer-Verlag, 1991.
14. U. M. Maurer and Y. Yacobi. A non-interactive public-key distribution system. *Designs, Codes and Cryptography*, 1996, 9: 305-316.
15. K. Nyberg and R.A. Rueppel. Message recovery for signature schemes based on the discrete logarithm problem. *Designs, Codes and Cryptography*, 1996, 7(1-2): 61-81.
16. S.C. Pohlig and M.E. Hellman. An improved algorithm for computing logarithms over $GF(P)$ and its cryptographic significance. *IEEE Trans. Inform. Theory*, 1978, 24: 106-110.
17. J.M. Pollard. Theorems on factorization and primality testing. *Proc. Cambridge Philos. Soc.*, 1974, 76: 521-528.
18. S. Popescu. A modification of the Tseng-Jan group signature scheme. *Studia Univ. Babes-Bolyai, Informatica*, 2000, XLV(2): 36-40. `http://www.cs.ubbcluj.ro/ ~studia-i/2000-2/` or `http://citeseer.nj.nec.com/504016.html`.
19. S. Saeednia. On the security of a convertible group signature schemes. *Information Processing Letters*, 2000, 73: 93-96.
20. A. Shamir. Identity-based cryptosystem based on the discrete logarithm problem. In: *Crypto'84, LNCS 196*, pp. 47-53. Springer-Verlag, 1985.
21. Shi Rong-Hua. An efficient secure group signature scheme. In: *Proc. of TENCON'02*, pp. 109-112. IEEE Computer Society, 2002.
22. D.X. Song. Practical forward secure group signature schemes. In: *Proceedings of the 8th ACM conference on Computer and Communications Security* (*CCS'01*), pp. 225-234. New York: ACM press, 2001.
23. H. Sun. Comment: improved group signature scheme based on discrete logarithm problem. *Electronics Letters*, 1999, 35(13): 1323-1324.
24. Y.-M. Tseng, and J.-K. Jan. Improved group signature based on discrete logarithm problem. *Electronics Letters*, 1999, 35(1): 37-38.
25. Y.-M. Tseng, and J.-K. Jan. Reply: improved group signature scheme based on discrete logarithm problem. *Electronics Letters*, 1999, 35(20): 1324.
26. Y.-M. Tseng, and J.-K. Jan. A novel ID-based group signature. In: T.L. Hwang and A.K. Lenstra, editors, *1998 International Computer Symposium, Workshop on Cryptology and Information Security*, Tainan, 1998, pp. 159-164.
27. Y.-M. Tseng, and J.-K. Jan. A novel ID-based group signature. *Information Sciences*, 1999, 120: 131-141. Elsevier Science.
28. C.-H. Wang, T. Hwang, and N.-Y. Lee. Comments on two group signatures. *Information Processing Letters*, 1999, 69: 95-97.
29. S. Xia, and J. You. A group signature scheme with strong separability. *The Journal of Systems and Software*, 2002, 60(3): 177-182. Elsevier Science.

30. F. Zhang and K. Kim. Cryptanalysis of two new signature schemes. Available at http://eprint.iacr.org/2002/167.
31. J. Zhang, J.-L. Wang, and Y. Wang. Two attacks on Xia-You group signature. Available at http://eprint.iacr.org/2002/177.
32. F. Zhang and K. Kim. Security of a new group signature scheme from IEEE TENCON'02. *Technical Reports 2003*, CAIS Lab.