Analysis on Popscu's Group Signature Scheme for Large Groups

Hyungki Choi and Kwangjo Kim

International Research Center for Information Security

Information and Communication Univ. (ICU), Korea.

Abstract

At SIC 2001, Popescu proposed an efficient group signature scheme for large groups [1]. However, this paper shows that his scheme is to be insecure by presenting a signature forgery. Using our attack, anyone (not necessarily a group member) can forge a signature on a message m, and since the attacker doesn't have to be the group member, the revocation manager cannot reveal the identity of the signer. Additionally, we modify Popescue's scheme to prevent the forgeary.

I. Introduction

In 1991, Chaum and van Heyst proposed the concept of a group signature scheme [2]. A group signature scheme allows a group member to sign messages anonymously on behalf of the group. More specifically, signatures can be verified with respect to a single public key of the group and do not reveal the identity of the signer. Futhermore, it must be infeasible to decide whether two signatures have been issued by the same group member. However, there exists a designated group manager who can, in case of later dispute, reveal the identity of the signer.

Group signatures can be divided into two parts: those have signature size linear to the number of group members, and those have fixed signature size such as Ateniese *et al.* [3]. In SIC 2001, Popescu [1] proposed an efficient group signature scheme for large groups that has the length of the group's public key and signatures does not depend on the size of the group. In this paper, we propose attack on Popescu's group signature scheme. In our attack, anyone (not necessarily a group member) can forge a signature on a message m, and since the attacker doesn't have to be the group member, the revocation manager cannot reveal the identity of the signer.

The rest of this paper is organized as follows: the next section reviews Popescu's group signature scheme in brief using the same notation as [1]. In Section III, we present how anyone can forge a signature on a message m. In Section IV, we modify Popescue's group signature scheme to prevent the forgeary, Finally, we conclude our paper in Section V.

II. Popscue's Group Signature Scheme

First, we show Popecue's group signature scheme briefly using the same notation as [1]. We omit the definition, assumption, and security proves etc. For further details, refer the original paper.

Like any group signature scheme, Popescue's scheme consists of Setup, Join, Sign, Verify, and Open. We review each phases as follows:

1. Setup

- The group manager chooses 2 random primes p', q' and computes $p=2p'+1$, $q=2q'+1$. Then, the group manager computes $n=pq$. Let l_n denote the bit-length of n, and chooses a public exponent $e>4$ such that e is relatively prime to $\Theta(n)$.
- The group manager selects g an element of Z_n^* of order n, and chooses an element $C\in Z_n^*$, a secret value $x\in Z_n^*$, and computes $y=g^x \pmod{n}$.
- Public key is $P=(n,e,g,y,h,C,l_n,\varepsilon, l_1,l_2)$, and secret key is $S=(p',q',x)$.

2. Join

Suppose now that a user wants to join the group. We assume that communication between a group member and the group manager is secure, *i.e.* private and authentic. A membership certificate in the group signature scheme consists of a pair of integers (X,δ) satisfying $X^e \equiv C+\delta \pmod{n}$ and $\delta\in[2^{l_1},2^{l_1}+2^{l_2}-1]$. To obtain his membership certificate, each user U_i must perform the following protocol with the group manager.

- The user U_i selects a random element $x_i\in[2^{l_1},2^{l_1}+2^{l_2}-1]$ and computes $ID_i=g^{x_i} \pmod{n}$, and must prove to the group manager that he knows $Dlog_g ID_i$ and that this value is in the inteval $(2^{l_1}-2^{\varepsilon(l_2+k)},2^{l_1}+2^{\varepsilon(l_2+k)+1})$.
- Then, the user U_i chooses a random number $r\in Z_n^*$ and computes $z=r^e(C+x_i) \pmod{n}$. He sends z to the group manager.
- The group manager computes $v=z^{1/e} \pmod{n} = r(C+x_i)^{1/e} \pmod{n}$ and sends v to the user U_i.
- The user U_i computes $A_i=v/r=(C+x_i)^{1/e} \pmod{n}$. The pair (A_i,x_i) is the membership certificate of the user U_i.

3. Sign

A group member U_i, with a membership certificate (A_i,x_i), can generate anonymous and unlikable group signatures on a message m as follows:

- Choose an integer $w\in_R\{0,1\}_2^*$ and compute $A=A_ih^w \pmod{n}$, $B=g^w \pmod{n}$, and $D=g_i^x y^w \pmod{n}$.
- Choose $r_1,r_2,r_3,r_4,$ and r_5 that satisfy the conditions shown in the paper [1]. Then, compute $d_1=B^{r_1}/g^{r_2} \pmod{n}$, $d_2=g^{x_i^2}D^{r_4}/y^{r_5} (\mathrm{mod}\, n)$, $d_3=g^{r_3} \pmod{n}$, and $d_4=g^{r_1}y^{r_3} \pmod{n}$.
- Compute $c=H(m\|g\|h\|y\|A\|B\|D\|d_1\|d_2\|d_3\|d_4)$.
- Compute $s_1=r_1-c(x_i-2^{l_1})$, $s_2=r_2-cx_iw$, $s_3=r_3-cw$, $s_4=r_4+x_i+c2^{l_1}$, and $c_5=r_5+x_iw+c2^{l_1}$.
- Send the group signature $(c,s_1,s_2,s_3,s_4,s_5,A,B,D)$ to the verifier.

4. Verify

The resulting signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ of a message m can be verified as follows:

- Compute $c' = H(m\|g\|h\|y\|A\|B\|D\| B^{s_1 - c2^{l_1}}/g^{s_2} (\bmod\ n)\|D^{s_4 - c2^{l_1}}/y^{s_5 - c2^{l_1}} y^{s_3} \|B^C g^{s_3} (\bmod\ n)\|D^c g^{s_1 - c2^{l_1}} y^{s_3})$
- Accept the group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ if and only if $c = c'$ and $s_1 \in \{-2^{l_2+k}, \ldots, 2^{\varepsilon(l_2+k)}\}$, $s_2 \in \{2^{l_G + l_1 + k}, \ldots, 2^{\varepsilon(l_G + l_1 + k)}\}$, $s_3 \in \{2^{l_G + k}, \ldots, 2^{\varepsilon(l_G + k)}\}$, $s_4 \in \{2^{l_2+k}, \ldots, 2^{\varepsilon(l_2+k)}\}$, and $s_5 \in \{-2^{l_2+k}, \ldots, 2^{\varepsilon(l_2+k)}\}$.

5. Open

Given a group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ the group manager can, by checking its correctness, find out which one of the group members issued this signature. He gives up if the signature is not correct. Otherwise, he performs the following steps:

- Recover ID_i (the identity of the user U_i) as $ID_i = D/B^x (\bmod\ n)$.
- Prove that $D\log_g y = D\log_B (D/ID_i \bmod\ n)$.

III. Attack on Popescu's Group Signature Scheme

In this section, we give attack on Popescu's group signature scheme. In our attack, anyone (who doesn't have to be a group memeber) can forge a signature on a message m. We describe our attack in detail as follows:

1) Let's denote an attacker ADV. ADV selects a random number $x \in_R Z_n^*$ and computes $ID = g^x (\bmod\ n)$. (x works as x_i to the group member).

2) As in Sign phase, ADV randomly choose $w \in_R \{0,1\}^{l_2}$ and $A \in Z_n^*$ (note that A is randomly chosen), and compute $B = g^w (\bmod\ n)$, and $D = g^x y^w (\bmod\ n)$.

3) Randomly Choose r_1, r_2, r_3, r_4, and r_5 that satisfy the conditions in the original scheme. Then, computes $d_1 = B^{r_1}/g^{r_2} (\bmod\ n)$, $d_2 = g^{x^2} D^{r_4}/y^{r_5} (\bmod \mathrm{n})$, $d_3 = g^{r_3} (\bmod\ n)$, and $d_4 = g^{r_1} y^{r_3} (\bmod\ n)$.

4) Compute $c = H(m\|g\|h\|y\|A\|B\|D\|d_1\| d_2\|d_3\|d_4)$.

5) Compute $s_1 = r_1 - c(x - 2^{l_1})$, $s_2 = r_2 - cxw$, $s_3 = r_3 - cw$, $s_4 = r_4 + x + c2^{l_1}$, and $s_5 = r_5 + xw + c2^{l_1}$.

Now, we created the signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$. If the verifier uses Verify phase and this signature is accepted, it means that we are able to forge the signature. Since s_1, s_2, s_3, s_4, and s_5 are the same except that we use x instead of x_i. We only need to check whether $c = c'$ is satisfied.

Since only difference c' from c is d_1, d_2, d_3, and d_4, we show whether followings are the same:

$$d_1 = B^{r_1}/g^{r_2} \overset{?}{=} B^{s_1 - c2^{l_1}}/g^{s_2}$$

$$= B^{r_1 - cx} / g^{r_2 - cxw}$$

$$= g^{wr_1} / g^{r_2}$$

$$= B^{r_1} / g^{r_2}$$

$$d_2 = g^{x^2} D^{r_4} / y^{r_5} \overset{?}{=} D^{s_4 - c2^{l_1}} / y^{s_5 - c2^{l_1}}$$

$$= D^{r_4 + x} / y^{r_5 + xw}$$

$$= (g^x y^w)^{r_4} / y^{r_5}$$

$$= D^{r_4} / y^{r_5}$$

$$d_3 = g^{r_3} \overset{?}{=} B^c g^{s_3}$$

$$= g^{wc} g^{r_3 - cw} = g^{r_3}$$

$$d_4 = g^{r_1} y^{r_3} \overset{?}{=} D^c g^{s_1 - c2^{l_1}} y^{s_3}$$

$$= (g^x y^w)^c g^{r_1 - cw} y^{r_3 - cw}$$

$$= g_q^r y^{r_3}$$

As we have shown the above, anyone can forge the signature. In addition, since an attacker randomly chooses x, even the revocation manager cannot trace who the actual signer was.

IV. Modification of Popescue Scheme

The problem of Popescue's group signature scheme is that during the Sign phase the user's private information is not used to generate the group signature. Thus, anonyone is able to forge the signature. In this section, we modified his original scheme to prevent the forgeary as follows.: in Sign phase, change d_1 as $d_1 = A^{r_1} A_i^{cx_i} / h^{r_2} (\text{mod}\ n)$. Then, the rest of the Sign phase is same as the original scheme. During Verify phase, we compute

$$c' = H(m \| g \| h \| y \| A \| B \| D \| A^{s_1 - c2^{l_1} / h^{x_2}} (\text{mod}\ n) \| D^{s_4 - c2^{l_1}} / y^{s_5 - c2^{l_1}} y^{s_3} \| B^C g^{s_3} (\text{mod}\ n) \| D^c g^{s_1 - c2^{l_1}} y^{s_3}).$$

Since all other parts are same as the original scheme, we only check whether $d_1 = A^{s_1 - c2^{l_1} / h^{x_2}} (\text{mod}\ n)$.

$$A^{s_1 - c2^{l_1} / h^{x_2}} (\text{mod}\ n) = A^{r_1 - cx_i} / h^{r_2 - cx_i w}$$

Since ($A = A_i h^w (\text{mod}\ n)$), we can rewrite $A^{r_1 - cx_i} / h^{r_2 - cx_i w}$ as $A^{r_1} A_i^{cx_i} / h^{r_2}$. In modified scheme, during the Sign phase the user must use its membership certificate to generate signature. Thus, only the user who knows the private information can generate the valid signature.

V. Concluding Remarks

In this letter, we have shown that the problem of Popescu's group signature scheme is that anyone (not necessarily the actual group member) can forge a signature on a message m. In addition, since an adversary selects x that randomly chosen by the adversary, if there is a problem occurred, revocation manger cannot identify the real signer of signature on a message. Then, we modify Popescue's scheme to prevent the forgeary. As a further work, we need more rigourous security proof on the modify Popescue scheme

References

[1] C. Popescu, ''An Efficient Group Signature Scheme for Large Groups'' *Studies in Informatics and Control Journal*, vol. 10, Number 1, 2001.

[2] D. Chaum and E. van. Heyst, ''Group Signatures'' *EUROCRYPT 1991*, LNCS 547, pp. 257-265, Springer-Verlag, 1991.

[3] G. Ateniese, J. Camenisch, M Joye, and G. Tsudik, ''A Practical and Provably Secure Coalition-Resistant Group Signature Scheme'' *Crypto 2000*, LNCS 1880, pp. 255-270, Plenum Press, 2000.