Efficient Membership Revocation in ACJT Group Signature

Jing-Liang Zhang and Yu-Min Wang

Abstract—How to find efficient and secure membership revocation algorithms is one of the most important issues standing in the way of real-world applications of group signatures. In this paper, the proof of knowledge of divisibility is given and a novel membership revocation method in ACJT group signature scheme is proposed: the group manager issues the product of the public keys of current members in the group, when a group member wants to sign, he should not only proves that he has a membership certificate, but also proves that the public key in his certificate divides exactly the public key product with zero knowledge. The proposed method is efficient since the group manager only needs one division and one exponentiation when a group member is deleted, while the signing and verifying procedure are independent of the number of current and excluded members, as well as the original group public key and membership certificates needn’t be changed.

Index Terms—ACJT group signature, dynamic group signature, information security, membership revocation, signature of knowledge.

1. Introduction

A group signature scheme[1][5] allows a group member to anonymously sign a message on behalf of a group, a verifier can be convinced that the signature is signed by some member in the group, but cannot find out which member provided it. On the other hand, a group signature has traceability: the group manager can open the signature and identify the identity of the signer if necessary. These salient features of group signatures make it attractive for many special applications, such as bidding, electronic cash and electronic payment.

However, a group is dynamic in practice, that is, the number of the group may increase or decrease. The join procedure of a group member is an inherent part of a group signature, but the early group signature schemes have no function of revocation, so how to find efficient and secure membership revocation algorithms is one of the most important issues standing in the way of real-world applications of group signatures.

The main contribution of this paper is to propose a novel revocation method in ACJT group signature[1], which is obtained by incorporating a new building block, i.e. a protocol for proving knowledge of divisibility. Compared with the prior revocation schemes[6][15], our scheme is more efficient and practical: the group manager only needs one division and one exponentiation, where the length of exponent is fixed, to update the public key for excluding a group member; the signing and verifying procedures are independent of the number of current and excluded members; the group member needn’t communicate with the group manager to update his membership certificates and can still use the original one.

2. Preliminaries

2.1 Number-Theoretic Assumptions

Let $G =< g >$ be a cyclic group whose order $\#G$ is unknown and its bit length $l_g$ is publicly known. There also exists a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ and a security parameter $\varepsilon > 1$.

Assumption 1. (Strong-RSA Assumption[1]) There exists a probabilistic algorithm $T$ such that for all probabilistic polynomial-time algorithms $A$, all polynomials $P(\cdot)$, and all sufficiently large $l_g$,

$$P_z\{z = u^e | (G, z) := T(l_g), (u, e > 1) := A(G, z) \} = 1/P(l_g).$$

Assumption 2. (Decisional Diffie-Hellman Assumption[1]) There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between the distributions $D$ and $R$, where $D = (g, g^x, g^y, g^{xy})$ and $R = (g, g^x, g^y, g^{xy})$ with $x, y, z \in Z_{q\ell_g}$.

2.2 ACJT Group Signature Scheme

We briefly introduce the ACJT group signature scheme[4].

Setup: The group manager (GM) chooses random secret $l_p$ bit primes $p', q'$ such that $p = 2p' + 1, q = 2q' + 1$ are prime, sets the modulus $n = pq$. Also, GM chooses random elements $a, a_0, g, h \in Z_{q\ell_g}$ and a random secret element $x \in Z_{\ell_g}$, sets $y = g^x \mod n$. Then, the group public key is $Y = (n, a, a_0, y, g, h)$ and the corresponding secret key is $S = (p', q', x)$.

Index Terms—ACJT group signature, dynamic group signature, information security, membership revocation, signature of knowledge.

1. Introduction

A group signature scheme[1][5] allows a group member to anonymously sign a message on behalf of a group, a verifier can be convinced that the signature is signed by some member in the group, but cannot find out which member provided it. On the other hand, a group signature has traceability: the group manager can open the signature and identify the identity of the signer if necessary. These salient features of group signatures make it attractive for many special applications, such as bidding, electronic cash and electronic payment.

However, a group is dynamic in practice, that is, the number of the group may increase or decrease. The join procedure of a group member is an inherent part of a group signature, but the early group signature schemes have no function of revocation, so how to find efficient and secure membership revocation algorithms is one of the most important issues standing in the way of real-world applications of group signatures.

The main contribution of this paper is to propose a novel revocation method in ACJT group signature[1], which is obtained by incorporating a new building block, i.e. a protocol for proving knowledge of divisibility. Compared with the prior revocation schemes[6][15], our scheme is more efficient and practical: the group manager only needs one division and one exponentiation, where the length of exponent is fixed, to update the public key for excluding a group member; the signing and verifying procedures are independent of the number of current and excluded members; the group member needn’t communicate with the group manager to update his membership certificates and can still use the original one.

2. Preliminaries

2.1 Number-Theoretic Assumptions

Let $G =< g >$ be a cyclic group whose order $\#G$ is unknown and its bit length $l_g$ is publicly known. There also exists a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ and a security parameter $\varepsilon > 1$.

Assumption 1. (Strong-RSA Assumption[1]) There exists a probabilistic algorithm $T$ such that for all probabilistic polynomial-time algorithms $A$, all polynomials $P(\cdot)$, and all sufficiently large $l_g$,

$$P_z\{z = u^e | (G, z) := T(l_g), (u, e > 1) := A(G, z) \} = 1/P(l_g).$$

Assumption 2. (Decisional Diffie-Hellman Assumption[1]) There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between the distributions $D$ and $R$, where $D = (g, g^x, g^y, g^{xy})$ and $R = (g, g^x, g^y, g^{xy})$ with $x, y, z \in Z_{q\ell_g}$.

2.2 ACJT Group Signature Scheme

We briefly introduce the ACJT group signature scheme[4].

Setup: The group manager (GM) chooses random secret $l_p$ bit primes $p', q'$ such that $p = 2p' + 1, q = 2q' + 1$ are prime, sets the modulus $n = pq$. Also, GM chooses random elements $a, a_0, g, h \in Z_{q\ell_g}$ and a random secret element $x \in Z_{\ell_g}$, sets $y = g^x \mod n$. Then, the group public key is $Y = (n, a, a_0, y, g, h)$ and the corresponding secret key is $S = (p', q', x)$. 

Index Terms—ACJT group signature, dynamic group signature, information security, membership revocation, signature of knowledge.

1. Introduction

A group signature scheme[1][5] allows a group member to anonymously sign a message on behalf of a group, a verifier can be convinced that the signature is signed by some member in the group, but cannot find out which member provided it. On the other hand, a group signature has traceability: the group manager can open the signature and identify the identity of the signer if necessary. These salient features of group signatures make it attractive for many special applications, such as bidding, electronic cash and electronic payment.

However, a group is dynamic in practice, that is, the number of the group may increase or decrease. The join procedure of a group member is an inherent part of a group signature, but the early group signature schemes have no function of revocation, so how to find efficient and secure membership revocation algorithms is one of the most important issues standing in the way of real-world applications of group signatures.

The main contribution of this paper is to propose a novel revocation method in ACJT group signature[1], which is obtained by incorporating a new building block, i.e. a protocol for proving knowledge of divisibility. Compared with the prior revocation schemes[6][15], our scheme is more efficient and practical: the group manager only needs one division and one exponentiation, where the length of exponent is fixed, to update the public key for excluding a group member; the signing and verifying procedures are independent of the number of current and excluded members; the group member needn’t communicate with the group manager to update his membership certificates and can still use the original one.

2. Preliminaries

2.1 Number-Theoretic Assumptions

Let $G =< g >$ be a cyclic group whose order $\#G$ is unknown and its bit length $l_g$ is publicly known. There also exists a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ and a security parameter $\varepsilon > 1$.

Assumption 1. (Strong-RSA Assumption[1]) There exists a probabilistic algorithm $T$ such that for all probabilistic polynomial-time algorithms $A$, all polynomials $P(\cdot)$, and all sufficiently large $l_g$,

$$P_z\{z = u^e | (G, z) := T(l_g), (u, e > 1) := A(G, z) \} = 1/P(l_g).$$

Assumption 2. (Decisional Diffie-Hellman Assumption[1]) There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between the distributions $D$ and $R$, where $D = (g, g^x, g^y, g^{xy})$ and $R = (g, g^x, g^y, g^{xy})$ with $x, y, z \in Z_{q\ell_g}$.

2.2 ACJT Group Signature Scheme

We briefly introduce the ACJT group signature scheme[4].

Setup: The group manager (GM) chooses random secret $l_p$ bit primes $p', q'$ such that $p = 2p' + 1, q = 2q' + 1$ are prime, sets the modulus $n = pq$. Also, GM chooses random elements $a, a_0, g, h \in Z_{q\ell_g}$ and a random secret element $x \in Z_{\ell_g}$, sets $y = g^x \mod n$. Then, the group public key is $Y = (n, a, a_0, y, g, h)$ and the corresponding secret key is $S = (p', q', x)$. 

Index Terms—ACJT group signature, dynamic group signature, information security, membership revocation, signature of knowledge.
Join: User $P_1$ interacts with GM and obtains his membership certificate $[A_i, e_i]$, where $e_i$ is a prime and $A_i$ is $e_i$th root of a number generated by GM and $P_1$.

Sign: $P_i$ selects $\omega \in_k [0,1]^{2^\beta}$, and computes
\[
T_1 = A_i y^\omega \mod n, \quad T_2 = g_\omega \mod n, \quad T_3 = g_\omega^e h_\omega \mod n,
\]
$P_i$ chooses $r_1, r_2, r_3, r_4$ and then computes:

A. $d_i = T_1^i / (a_i^r y^s), \quad d_4 = g_\omega^r, \quad d_3 = g_\omega^s, \quad d_1 = g_\omega^h$.

B. $c = H(g, h, y, a_i, o_i, A_i, T_2, T_3, d_1, d_2, d_3, d_4, m)$.

C. $s_1 = r_1 - c(e_i - 2^\beta), \quad s_2 = r_2 - c(x_i - 2^\beta), \quad s_3 = r_3 - ce_i, \omega, \quad s_4 = r_4 - c e_\omega$.

Then the signature is $(c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$.

Verify: A verifier can check the correctness of the signature.

Open: GM can find the signer from the signature $(c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$.

3. Achieving Revocations

3.1 Proving Divisibility

Suppose that there exists a group $G = \langle g, h \rangle$ of order $\phi(n)$ and a publicly number $E$. If a prover, who does not know $\phi(n)$ and $\log_2 h$, wants to prove that he knows a number $e$ satisfying $e \mid E$ without leaking any information about $e$, he may use the following protocol:

**Prover:**

A. Randomly choose $o_1, o_2, o_3$ and computes:
\[
y_1 = g_1^{o_1}, \quad y_2 = g_1^{o_2}, \quad y_3 = g_1^{o_3}, \quad y_4 = h^{(a o_1 + o_3)}
\]
\[
T_1 = g_1^{h y_1}, \quad T_2 = g_1^{h y_2}, \quad T_3 = (T_1)^{y_3} h^{y_4}.
\]

B. Randomly chooses $r_1, r_2, r_3, r_4, r_5, r_6$ and computes:
\[
R_1 = g_1^{r_1}, \quad R_2 = g_1^{r_2}, \quad R_3 = g_1^{r_3}, \quad R_4 = h^{r_4}
\]
\[
R_5 = g_1^{h R_1}, \quad R_6 = g_1^{h R_2}, \quad R_7 = T_3^2 h^{R_6}
\]

C. Computes
\[
c = H(g, h, E, y_1, y_2, y_3, y_4, T_1, T_2, T_3, T_4, T_5, o_1, o_2, o_3, R_1, R_2, R_3, R_4, R_5, R_6, R_7)
\]
\[
s_1 = r_1 - c o_1, \quad s_2 = r_2 - c o_2, \quad s_3 = r_3 - c o_3, \quad s_4 = r_4 - c e_1, \quad s_5 = r_5 - c e_2, \quad s_6 = r_6 - c e_3
\]

D. Publishes $(y_1, y_2, y_3, y_4, T_1, T_2, T_3, c, s_1, s_2, s_3, s_4, s_5, s_6)$.

**Verifier:**

A. Computes
\[
c' = H(g, h, E, y_1, y_2, y_3, y_4, T_1, T_2, T_3, T_4, T_5, T_6, T_7)
\]
\[
g_1^{n y_1 c} g_1^{y_2 c} g_1^{y_3 c} g_1^{y_4 c} h_1^{y_1}, \quad g_1^{y_2 h} T_1^{c}, \quad g_1^{y_3 h} T_2^{c}, \quad g_1^{y_4 h} T_3^{c}
\]

B. Verifies $c' = c$ and $T_7 y_4 = g_1^e$, then $e \mid E$; otherwise, the proof fails.

In above protocol, the value $E$ is an integer and may be very large, so if we apply the above method directly in membership revocation, the computation of $g^e$ may become inefficient. In order to achieve the efficient membership revocation, we will use $g_\omega = g_1^{E \mod \phi(n)}$ instead of $g^e$ in the above protocol. Here, $g_\omega$ is a public value computed by $\phi(n)$.

Using the notation introduced by Camenisch and Lysanskaia[60], we get the new protocol for proving the knowledge of divisibility, denoted by

$PK1(\alpha, \beta, \theta, \tau, \xi, \zeta): y_1 = g_\omega^\theta, \quad y_2 = g_\omega^\tau, \quad y_3 = g_\omega^\xi, \quad y_4 = h_\omega^\zeta$

$\land T_1 = g_\omega^h T_2 = g_\omega^{h'} T_3 = T_1^{\theta} T_2^{\tau} T_3^{\zeta}$.

**Theorem 1.** Suppose $E$ is relatively prime to $\phi(n)$. Under the strong RSA assumptions, the interactive protocol corresponding to above $PK1$ protocol is a knowledge proof of the fact that the prover has a number $\alpha$ which is a divisor of $E$.

**Proof:** We have to show that the knowledge extractor is able to recover $\alpha$ which is a divisor of $E$ once it has found two accepting tupsles as follows:

\[
(y_1, y_2, y_3, y_4, T_1, T_2, T_3, c, s_1, s_2, s_3, s_4, s_5, s_6)
\]
\[
(y_1, y_2, y_3, y_4, T_1, T_2, T_3, c, s_1, s_2, s_3, s_4, s_5, s_6)
\]

Because $R_1 = g_1^{y_1 c} g_1^{y_2 c} g_1^{y_3 c} g_1^{y_4 c}$, we have $y_1 c e_c = g_1^{y_1 c}$.

Let $d_i = \gcd(c - c', s_i' - s_i)$, by using the extended Euclidean algorithm, we can obtain the values $u_i, v_i$, such that
\[
u_i (c - c') + v_i (s_i' - s_i) = d_i,
\]
and hence we have
\[
g = g_1^{u_i (c - c') + v_i (s_i' - s_i) / d_i} = (g_1^{y_i})^{(c - c') / d_i}
\]
If $d_i | (c - c')$, then $g_1^{y_i}$ is a $(c - c') / d_i$ root of $g$, this contradicts the strong RSA assumption. So $d_i | (c - c')$ and we can compute the integer $\theta = (s_i' - s_i) / (c - c')$, such that
\[
y_4 = g_1^{\theta}
\]
Likewise, from $R_3 = g_1^{y_3 c} T_3 = g_1^{y_3 c} h_1^{T_3 c}$, we obtain $\alpha = (s_i' - s_i) / (c - c')$, such that $T_1 = g_1^{T_1 h^{T_1 c}}$.

From $R_3 = g_1^{y_3 c} y_3 = g_1^{y_3 c} y_3 = h_1^{y_3 c}, \quad R_1 = h_1^{y_3 c} y_3 = h_1^{y_3 c} y_3 = h_1^{y_3 c} y_3$, we can obtain $\xi = (s_i' - s_i) / (c - c'), \quad \zeta = (s_i' - s_i) / (c - c'), \quad \zeta = (s_i' - s_i) / (c - c')$, such that $y_1 = g_1^{\xi}, \quad y_3 = h_1^{\zeta}, \quad T_1 = (T_1 h^{T_1 c})$. By $g_1^{E / \phi(n)} = g_1^{E / \phi(n)}$, we get $g_1^{E / \phi(n)} h_1^{T_1 c + c'}$. Because the prover doesn’t know the discrete logarithm of $h$ to the base $g$, there must be $g_1^{E / \phi(n)} h_1^{T_1 c + c'} = 1, \quad \beta + \xi + \zeta = 0$. Otherwise, the prover can calculate the discrete logarithm of $h$ to the
base $g$, i.e., $h = g^{(E \mod \phi(n) - a \beta)(j \beta + \zeta)}$, this contradicts the discrete logarithm assumption. That is, $E = a \beta \mod \phi(n)$. But $E$ is relatively prime to $\phi(n)$, so $a \beta / E = 1 \mod \phi(n)$.

Let $d = \gcd(\alpha, E)$, if $d \neq \alpha$, then
\[
\frac{(\alpha/d)\beta}{(E/d)} = \frac{1}{\alpha} \mod \phi(n).
\]
By $d = \gcd(\alpha, E)$, $\alpha/d$ is relatively prime to $E/d$, so
\[
(\alpha/d)\beta (E/d) = (\alpha/d)(\beta \mod E) = 1 \mod \phi(n).
\]
Hence, let $s = \beta d / E$, then $(\alpha/d)s = 1 \mod \phi(n)$, i.e., for the integer $(\alpha/d)\beta = 1 \mod \phi(n)$.

This contradicts the Assumption 1, so $d = \alpha$, i.e., $\alpha$ is a divisor of $E$.

### 3.2 Application in Revocation of Membership

In the new scheme, the setup, join and open procedures are the same as ACJT scheme. So we just describe the revoke, sign, verify and procedures as follows.

**Revoke:** Suppose that the set of group members is $G = \{G_1, G_2, \ldots, G_n\}$, the prime of the member $G_i$ in his membership certificate is $e_i$. GM computes $E = e_1 e_2 \cdots e_m$. If the set of excluded members is $G' = \{G_1, G_2, \ldots, G_n\}$, then GM updates $E$ with $E = (e_1 e_2 \cdots e_m) / (e_{\ell_1} e_{\ell_2} \cdots e_{\ell_\ell})$ and computes $g_E = g^{E \mod \phi(n)}$.

**Sign:** A group member can sign using the method in the original scheme, but he must prove that he is not excluded from the group, i.e., prove that the $e_i$ in his membership certificate $(A, e_i)$ is a divisor of $E$. The sign procedure can be obtained as follows:

A. Compute $a = E / e_i$

B. Choose $u, v, w \in \mathbb{Z} / 2^l$ and compute

\[
T_1 = A y^u, \quad T_2 = g^u, \quad T_3 = g^{v h^u}, \quad T_4 = g^v,
\]

\[
T_5 = g^{w h^u}, \quad T_6 = g^w, \quad T_7 = (T_5)^{t h^u}, \quad T_8 = h^{(a + w)}.
\]

C. Generate

$PK2(\alpha, \beta, \delta, e) : a_0 = T_1^u (1/a)^d (1/y)^d \wedge T_2 = g^c \wedge 1 = T_2^u (1/g)^d$

$\wedge T_6 = g^{w h^u} \wedge T_7 = (T_7) h^u$.

$PK3(\alpha, \eta, r, \xi, \zeta) : T_2 = g^c \wedge T_3 = g^f \wedge T_6 = g^s \wedge T_8 = h^5$

$\wedge T_7 = g^{w h^u} \wedge T_8 = (T_7) h^u$.

D. The signature is $(T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, PK2, PK3)$

The protocol $PK3$ is the same as the ACJT scheme, which is used to prove that the signer has membership certificate $(A, e_i, x_i)$, while $PK3$ is used to prove that the $e_i$ is a divisor of $E$.

**Verify:** Check the correctness of $PK2$ and $PK3$.

### 4. Security Analysis

**Lemma 1:** Under the strong RSA assumption, a group certificate $(A, e_i, x_i)$ can be generated only by the group manager provided that the number $K$ of certificates the group manager issues is polynomially bounded.

**Lemma 2:** Under the strong RSA assumption, the interactive protocol corresponding to $PK2$ is a statistical zero-knowledge proof of knowledge of $(A, e_i, x_i)$.

Our proposed scheme satisfies all security properties required by a group signature scheme.

**Correctness:** This is achieved by checking the correctness of $PK2$ and $PK3$.

**Unforgeability:** If the user is neither a group member nor an excluded group member, then from Lemma 1, he can’t get the membership certificate $(A, e_i, x_i)$. If the user is an excluded member, the $e_i$ in his original certificate $(A, e_i, x_i)$ is not a divisor of the updated $E$, and from Theorem 1, he can not produce an $e_i$ which is a divisor of $E$. Hence, only group members are able to sign messages on behalf of the group.

**Anonymity:** Given a valid group signature $(T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, PK2, PK3)$.

From Theorem 1 and Lemma 2, there is no information revealed from $PK2$ and $PK3$. Furthermore, it is infeasible to get $(A, e_i, x_i)$ from $(T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8)$ because computing discrete logarithms is assumed to be infeasible.

**Exculpability:** From Lemma 1, we know that a group member cannot sign on behalf of other group members. Next, the group manager cannot get $x_i$ from $a^x$ because of the difficulty of computing discrete logarithms, so he cannot sign on behalf of a group member.

**Traceability:** The group manager may get $A_i$ by computing $T_1^T / T_2^T$, and then identify the actual signer.

**Coalition-resistance:** This follows from Theorem 1 and Lemma 1.

**Unlinkability:** By Theorem 1 and Lemma 2, there is no information revealed from $PK2$ and $PK3$. Furthermore, $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8$ are unconditionally binding with random numbers $u, v$ and $w$, so it is computationally hard to decide whether two valid signatures are signed by the same group member. In the following, we show that even when the group member is excluded, i.e., his $e_i$ is published, other members can’t link his past signatures. Suppose that $(T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8)$ and

$(T_1', T_2', T_3', T_4', T_5', T_6', T_7', T_8')$
are two signatures. When $e_i$ is published, other members can get $a = E/e_i$. If they want to know whether the two signatures are signed by the same member, they must decide whether the discrete logarithms

$$\log_g T_1/T'_1 = \log_g T_2/T'_2 = \log_g T_3/T'_3$$

or

$$\log_g T_4/T'_4 = \log_g T_5/T'_5$$

or

$$\log_g T_6/T'_6 = \log_g T_7/t'_7 = \log_g T_8/t'_8 = \log_g T_{9}/T_{9}^* .$$

This is impossible under the Decisional Diffie Hellman assumption.

**Revocability:** The excluded group member $P_i$ has two possible ways to pretend his membership: 1) he continues to use $e_i$ in his original certificate, but it is impossible by Theorem 1; 2) he chooses a new prime $e$ such that $e | E$ but it is also impossible from Lemma 1.

## 5. Conclusions

In this paper, we present a new revocation method for ACJT group signature scheme. The main idea is that the group manager publishes the product of $e_i$'s in the certificates corresponding to the non-excluded members and a legitimate group member should prove that his $e_i$ is a divisor of the product when he is signing. The new revocation scheme is efficient because the size of signature is constant and the group manager needs only one division and one exponentiation with the fixed length of exponent for excluding a member. Also, the signing and verifying procedures are independent of the number of current members and excluded members. Furthermore, the group members needn’t communicate frequently with the group manager to update their membership certificates, they can use their original certificates.

**References**


**Jing-Liang Zhang** was born in Shandong, China, in 1973. He received his M.S. degree from Department of Mathematics, Shannxi Normal University in 1999. He is now pursing his Ph. D. degree with State Key Laboratory of Integrated Service Networks, Xidian University. His research interests include cryptography and its applications.

**Yu-Min Wang** was born in Beijing, China, in 1936. He is a professor and Ph.D. supervisor with State Key Laboratory of Integrated Service Networks, Xidian University. His research interests focus on the areas of the general theory of communication, information theory, coding, cryptography and network security.