Mathematica Moravica
Vol. **7** (2003), 51–59

A METHOD FOR REVOCATION IN GROUP SIGNATURE SCHEMES

Constantin Popescu*

Abstract. A group signature scheme allows any group member to sign on behalf of the group in an anonymous and unlinkable fashion. In the event of a dispute, a designated trusted entity can reveal the identity of the signer. In this paper we propose a revocation method for group signatures based on the group signature scheme from [12]. This method requires no time periods and offers constant length signatures.

1. Introduction

Group signatures, introduced by D. Chaum and E. Heyst at Eurocrypt'91 [9], allow individual members of a group to sign messages on behalf of the group. Group signatures are publicly verifiable but anonymous in that, no one, with the exception of a designated group manager, can establish the identity of a signer. Furthermore, group signatures are unlinkable which makes it computationally hard to establish whether or not multiple signatures are produced by the same group member. At the same time, no one, including the group manager, can misattribute a valid group signature. A group signature scheme could for instance be used in many specialized applications, such as voting and bidding. They can, for example, be used in invitations to submit tenders. All companies submitting a tender form a group and each company signs its tender anonymously using the group signature. Once the preferred tender is selected, the winner can be traced while the other bidders

AMS (MOS) Subject Classification 1991. Primary: 54H25.

Key words and phrases: Group signature scheme, revocation of group members, Okamoto-Shiraishi assumption.

*The author is presently at "Centre for Quantifiable Quality of Service in Communication Systems" (Q2S), NTNU, Trondheim, Norway. The centre is appointed Centre of Excellence by The Research Council of Norway. It is financed by the Research Council, NTNU and UNINETT, and supported by Telenor.

remain anonymous. More generally, group signatures can be used to conceal organizational structures, e.g., when a company or a government agency issues a signed statement. Also, a group signature scheme could be used by an employee of a large company to sign documents on behalf of the company. A further application of a group signature scheme is electronic cash as was pointed out in [11]. In this case, several banks issue coins, but it is impossible for shops to find out which bank issued a coin that is obtained from a customer. The central bank plays the role of the group manager and all other banks issuing coins are group members.

A number of interesting group signature schemes have been proposed [1], [2], [3], [6], [8]. However, the schemes presented in [3], [5], [8], [12] support growing membership: new members can join without changes in the group public key or re-issuing group membership certificates for existing members. Shrinking group membership has not been given the same attention. In many realistic group settings, group members are equally likely to join, leave voluntarily or be excluded from the group. Bresson and Stern [4] proposed the first viable and elegant solution for revocation of group signatures. Unfortunately, their solution requires the signature size to be linear with respect to the number of revoked members. Moreover, it is based on the group signature scheme proposed by Camenisch and Stadler [5] which has been found later to have certain security problems. Last year, Song [13] proposed two interesting revocation methods based on the Ateniese et al. scheme [3]. Both methods are notable since they also provide retroactive revocation as well as forward security. Moreover, they offer constant length signatures which is an improvement over the Bresson and Stern's solution. However, one important feature of Song's methods is the use of fixed time periods to support revocation. In particular, each member's certificate must evolve in every time period and any and all verifiers must be aware of this evolution. Also, the maximum number of time periods is fixed and embedded in each member's group certificate. While appropriate for some settings, this solution is not very general since it is hard to revoke a member within a time period. Furthermore, the security of one of the methods is based on a new and perhaps uncertain cryptographic assumption which is appreciably stronger than the Decision Diffie-Hellman assumption. The second scheme relies on the existence of an efficient method of deterministically computing a fixed length sequence of prime numbers starting with an initial prime.

In this paper we construct a revocation method (for group signatures), that requires no time periods and offers constant length signatures, based on the group signature scheme from [12].

2. Preliminaries

Group signature schemes are typically defined as follows (see more details in [7]):

Definicija 2.1. *A group signature scheme is a digital signature scheme comprised of the following algorithms and protocols:*

1. ***Setup**: The public output is the group's public key P. The private outputs are the individual secret keys x_G for the each group member, the secret key x_M for the group manager.*
2. ***Join**: An interactive protocol between the group manager and a user that results in the user becoming a new group member.*
3. ***Sign**: An interactive protocol between the group member Alice and an external user, which on input message m from the user, the Alice's secret key x_G and the group's public key P outputs a group signature σ.*
4. ***Verify**: An algorithm that on input a message m, a signature σ and the group's public key P returns 1 if and only if σ was generated by any group member using the protocol **Sign** on input x_G, m and P.*
5. ***Open**: A tracing algorithm that on input a signature σ, a message m, the group manager's secret key x_M and the group's public key P returns the identity ID of the group member who issued the signature σ.*

A secure group signature scheme must satisfy the following properties:

1. Correctness: Signatures produced by a group member using **Sign** must be accepted by **Verify**.
2. Unforgeability: Only group members are able to sign messages on behalf of the group.
3. Anonimity: Given a valid signature, identifying the actual signer is computationally hard for everyone but the group manager.
4. Unlinkability: Deciding whether two different signatures were computed by the same group member is computationally hard.
5. Exculpability: Even if the group manager and some of the group members collude, they cannot sign on behalf of non-involved group members.
6. Traceability: The group manager can always establish the identity of the member who issued a valid signature. Therefore, any colluding subset of group members cannot generate a valid signature that the group manager cannot link to one of the colluding group members.

7. Revocability: A group signature produced using **Sign** by a revoked member must be rejected using the **Verify** algorithm. Equivalently, a group signature produced using **Sign** by a valid member must be accepted by **Verify**.

The efficiency of a group signature scheme depends on a number of factors. Usually, the costs of **Sign** and **Verify** as well as the sizes of the group signature and the group public key are the most important efficiency measures.

3. The Basic Group Signature Scheme

In this section we provide an overview of the group signature scheme from reference [12]. This group signature scheme is based on the Okamoto-Shiraishi assumption [10]. The symbol $\|$ denotes the concatenation of two binary string (or of the binary representation of group elements and integers).

3.1 Setup

The setup procedure is as follow. The group manager must perform the following steps:

1. Chooses random primes p', q' and computes the prime elements p and q such that $p = 2p' + 1$ and $q = 2q' + 1$. Then, the group manager computes an RSA-like modulus $n = pq$. Let l_n denotes the bit-length of n.
2. Chooses a public exponent $e > 4$ such that e is relatively prime to $\varphi(n)$.
3. Selects g an element of $\mathbb{Z}_n^*$ of order n. Let $G =< g >$ be a cyclic subgroup of $\mathbb{Z}_n^*$ of order l_G.
4. Selects an element $C \in \mathbb{Z}_n^*$ and an element $h \in G$ whose discrete logarithm to the base g must not be known.
5. Chooses a secret value $x \in \mathbb{Z}_n^*$ and computes $y = g^x (mod\ n)$.
6. Finally, a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ and security parameters $\epsilon > 1, l_1, l_2$ are set. An example for choosing the parameters $\epsilon, k, e, l_n, l_1, l_2, l_G$ is given in [12]. In [12], we proposed to use our group signature scheme with the following parameters: $l_n = 1200$, $e = 5$, $\epsilon = 9/8$, $k = 160$, $l_1 = 860$, $l_2 = 600$.

The public key is $P = (n, e, g, y, h, C, l_n, \epsilon, l_1, l_2, l_G, H, k)$ and the secret key is $S = (p', q', x)$. In practice, components of P must be verifiable to prevent framing attacks (e.g., see [10]). A membership certificate in our group signature scheme consists of a pair of integers (X, δ) satisfying $X^e \equiv C + \delta (mod\ n)$ and $\delta \in [2^{l_1}, 2^{l_1} + 2^{l_2} - 1]$.

3.2 Join

Suppose now that a user wants to join the group. We assume that communication between the group member and the group manager is secure, i.e., private and authentic. To obtain his membership certificate, each user U_i must perform the following protocol with the group manager.

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

Consequently, at the end of the protocol, the group manager does not know the membership certificate (A_i, x_i) of the user U_i. The group manager creates a new entry in the group database and stores ID_i in the new entry.

3.3 Sign

A group member U_i, with a membership certificate (A_i, x_i), can generate anonymous and unlinkable group signatures on a message $m \in \{0,1\}^*$ as follows.

1. Chooses a random integer $w \in \{0,1\}^{l_2}$ and computes
$$A = A_i h^w (mod\ n), \quad B = g^w (mod\ n),$$
$$D = g^{x_i} y^w (mod\ n).$$
2. Chooses random integers $r_1 \in \{0,1\}^{\epsilon(l_2+k)}$, $r_2 \in \{0,1\}^{\epsilon(l_G+l_1+k)}$, $r_3 \in \{0,1\}^{\epsilon(l_G+k)}$, $r_4 \in \{0,1\}^{\epsilon(l_2+k)}$, $r_5 \in \{0,1\}^{\epsilon(l_2+k)}$ and computes
$$d_1 = B^{r_1}/g^{r_2}(mod\ n)$$
$$d_2 = g^{x_i^2} D^{r_4}/y^{r_5}(mod\ n)$$
$$d_3 = g^{r_3}(mod\ n)$$
$$d_4 = g^{r_1} y^{r_3}(mod\ n).$$
3. Computes
$$c = H(m\|g\|h\|y\|A\|B\|D\|d_1\|d_2\|d_3\|d_4)$$
4. Computes $s_1 = r_1 - c(x_i - 2^{l_1})$, $s_2 = r_2 - cx_i w$, $s_3 = r_3 - cw$, $s_4 = r_4 + x_i + c2^{l_1}$, $s_5 = r_5 + x_i w + c2^{l_1}$ (in $\mathbb{Z}$).
5. Send the group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ to verifier.

3.4 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:

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

3.5 Open

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

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

4. Our Solution for Revocation in Group Signatures

We begin by assuming, as usual, that a Certificate Revocation List (CRL) is a structure available at all times from a number of well-known public repositories or servers. A CRL is also assumed to be signed and timestamped by its issuer which can be a universally trusted CA, a group manager or some other trusted party.

In addition to the usual components of a group signature scheme we introduce an additional algorithm called **Revoke**. Also, as can be expected, revocation influences **Sign** and **Verify** algorithms. The **Join** and **Open** components remain unchanged. The only change in **Setup** is as follows:

- (new step) Select $\overline{G} =< \overline{g} >$ of order n in which computing discrete logarithms is hard. For example, $\overline{G}$ can be a subgroup of $\mathbb{Z}_{\overline{p}}^*$ for a prime $\overline{p}$ such that n divide $(\overline{p} - 1)$.

The new **Revoke** algorithm shown below is executed by the group manager whenever a member or a collection of members leaves or is expelled.

Revoke:

We use s to denote the index of the current CRL issue and we assume that l users $U_1, ..., U_l$ are to be revoked:

1. Choose a random element $b_s \in QR(n)$ of order $p'q'$. This value b_s becomes the current revocation base.
2. For each revoked U_j, $1 \leq j \leq l$ compute
$$V_{s,j} = b_s^{x_j}.$$
3. The actual revocation list is then published
$$CRL_s = \{b_s, V_{s,j} \,|0 < j < l+1\}.$$

In the **Sign** algorithm, as part of step 1, member U_i generates two additional values:

$$\begin{aligned} E &= t = \overline{g}^r, where \quad r \in \mathbb{Z}_n^* \\ F &= t^{b_s^{x_i}} (mod\ n). \end{aligned}$$

The user U_i proves, in zero knowledge, that the double discrete logarithm of F with bases t and b_s, respectively is the same as the discrete logarithm of D's representation base $\overline{g}$ and y respectively. Since D is computed as $g^{x_i} y^w (mod\ n)$, the resulting proof of knowledge is verifiable if and only if the same x_i is used the construction of both F and D.

In the **Verify** algorithm we introduce a new steps 3 and 4:

3. For each $V_{s,j} \in CRL$, check if
$$F = E^{V_{s,j}} (mod\ n).$$
4. Check the proof of equality of double discrete logarithm of F and discrete logarithm of D's representation base $\overline{g}$.

The intuition behind this scheme is straight forward. If a member U_i is revoked, $V_{s,i}$ is published as part of the current group CRL. Thereafter, in order to produce a group signature, U_i needs to prove that $b_s^{e_i}$ does not appear on the CRL which is impossible since $b_s^{e_i} = V_{s,i}$ for some j if U_i is revoked.

We claim that our scheme provides backward unlinkability because signatures produced by a revoked user prior to revocation in earlier CRL epochs can not be linked to those produced after revocation. Suppose that an adversary is able to link a pre-revocation signature to a post-revocation signature. Then, she can only do so with the help of the new values E and F. Since $E = t$ is chosen at random for each signature, the only way the adversary can link two signatures is using $F = t^{b_s^{x_i}}$. However, this is impossible since the respective b_s values are different and unrelated for any pair of signatures computed in different CRL epochs. To be more specific, we need to consider two cases: linking two signatures from different CRL epochs and linking two signatures from the same CRL epoch. It is easy to see that the former is infeasible for some $F_1 = t'^{b_s^{e_i}}$ and $F_2 = t''^{b_s^{e_i}}$ where $t' \neq t''$, based on a well-known variant of the Decisional Diffie-Hellman problem.

5. Efficiency Considerations

Our revocation scheme is quasi-efficient in that a group signature is of fixed size and a signer performs a constant amount of work in generating a signature. This is, as claimed earlier, an improvement on prior results. However, proofs involving double discrete logarithms are notoriously expensive. For example, if we assume a hash function $H : \{0,1\}^* \rightarrow \{0,1\}^k$ where $k = 160$ bits, then each **Sign** operation will take approximately 500 exponentiations. The cost of **Verify** is roughly the same. Moreover, with a 1024 bit modulus, a signature can range into hundreds of Kbits.

Despite the usage of double discrete logarithm proofs, in contrast with Bresson and Stern's scheme [4], the cost of **Sign** in our scheme is constant (independent of group size of number of revoked members) and signatures are of a fixed size. Comparing with Song's schemes, our scheme is more expensive for both **Sign** and **Verify** due to the double discrete logarithms proof. One advantage of our scheme is in not using fixed (in length and number) time periods. Consequently, a new revocation list can be issued at any time. Also, we introduce no new cryptographic assumptions. Song's two schemes, however, have the benefit of retroactive public revocability meaning that a member's signatures can be revoked for one or more past time periods. This is a feature that our method does not possess.

The cost of **Revoke** in our scheme is linear in the number of revoked members. Group manager performs one exponentiation for CRL entry $V_{s,j}$. This is comparable with prior results in both [4] and [13] schemes.

6. Conclusion

In this paper we proposed a revocation method for group signatures based on the group signature scheme from reference [12]. Our method is more practical than prior art due to fixed size signatures and constant work by signers. On the other hand, it requires the use of proofs of knowledge involving double discrete logarithms which results in hundreds of exponentiations per signature.

7. References

[1] G. Ateniese, G. Tsudik, *Group signature a la carte*, Tenth Anual ACM-SIAM Symposium on Discrete Algorithms (SODA'99), 1999.

[2] G. Ateniese, G. Tsudik, *Some open issues and new directions in group signatures*, Financial Cryptography (FC'99), Lecture Notes in Computer Science, Springer-Verlag, 1999.

[3] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik, *A Practical and Provably Secure Coalition-Resistant Group Signature Scheme*, Advances in Cryptology - CRYPTO 2000, vol. 1880, Lecture Notes in Computer Science, Springer Verlag, pp. 255-270, 2000.

[4] E. Bresson, J. Stern, *Efficient revocation in Group Signatures*, Proceedings of Public Key Cryptography, Springer-Verlag, 2001.

[5] J. Camenisch, M. Stadler, *Efficient group signature schemes for large groups*, Advances in Cryptology-CRYPTO'97, Lecture Notes in Computer Science, Springer-Verlag, 1296, pp. 410-424, 1997.

[6] J. Camenisch, *Efficient and generalized group signatures*, Advances in Cryptology-EUROCRYPT'97, Lecture Notes in Computer Science, Springer-Verlag, 1233, pp. 465-479, 1997.

[7] J. Camenisch, *Group signature schemes and payment systems based on the discrete logarithm problem, PhD thesis*, Vol. 2, ETH Series in Information Security on Cryptography, Hartung-Gorre Verlag, 1998.

[8] J. Camenisch, M. Michels, *A group signature scheme with improved efficiency*, Advances in Cryptology-ASIACRYPT'98, Lecture Notes in Computer Science, Springer-Verlag, 1514, pp. 160-174, 1998.

[9] D. Chaum, E. van Heyst, *Group signatures*, Advances in Cryptology-EUROCRYPT'91, Lecture Notes in Computer Science, Springer-Verlag, 547, pp. 257-265, 1991.

[10] E. Fujisaki, T. Okamoto, *Statistical zero knowledge protocols to prove modular polynomial relations*, Advances in Cryptology-CRYPTO'97, Lecture Notes in Computer Science, Springer-Verlag, 1297, pp. 16-30, 1997.

[11] A. Lysyanskaya, Z. Ramzan, *Group blind signature: A scalable solution to electronic cash*, Financial Cryptography (FC'98), Lecture Notes in Computer Science, Springer-Verlag, 1465, pp. 184-197, 1998.

[12] C. Popescu, *Group signature schemes based on the difficulty of computation of approximate e-th roots*, Proceedings of Protocols for Multimedia Systems (PROMS 2000), Poland, pp. 325-331, 2000.

[13] D. Song, *Practical Forward-Secure Group Signature Schemes*, Proceedings of 2001 ACM Symposium on Computer and Communication Security, 2001.

University of Oradea
Department of Mathematics
Str. Armatei Romane 5, Oradea, Romania
E-mail: cpopescu@uoradea.ro

Received: January 20, 2003.