An Efficient Group Signature Scheme for Large Groups

Constantin Popescu

Department of Mathematics

University of Oradea

5 Armata Romana str.

Oradea

ROMANIA

E-mail: cpopescu@math.uoradea.ro

Abstract: A group signature allows a group member to sign messages anonymously on behalf of the group. Only can a designated entity determine the identity of the group member who issued a given signature. In this paper we propose a new group signature scheme, suitable for large groups, i.e. the length of the group's public key and of signatures does not depend on the size of the group. Our group signature scheme is based on the difficulty of computation of approximate *e*-th roots modulo a composite number and is more efficient than the previous ones.

Keywords: Group signature scheme, large groups, membership certificate, discrete logarithms, signature of knowledge

Constantin Popescu was born at Danesti, Romania, on 21st October , 1967. He received the MSc. degree in Computer Science from the University of Timisoara, Timisoara, Romania, in 1992. In 1992 he became an Assistant Professor at the Department of Mathematics, University of Oradea, Oradea, Romania. Since 1998 he has been Lecturer at the Department of Mathematics, University of Oradea. Since 1997 he has been a Ph. D student at the Babes-Bolyai University, Cluj Napoca. He has worked in the area of data security and privacy and published several papers in the field of cryptography. His current research interests include cryptography, network security, group signatures, identification schemes.

1. Introduction

A group signature allows any member of a group to sign on behalf of the group. Group signatures are publicly verifiable and can be verified with respect to a single group public key. Only a designated group manager can revoke the anonymity of a group signature and find out the identity of the group member who issued a given signature. 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. 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 pointed out in [15]. 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. Central bank plays the role of the group manager, with all the other banks issuing coins as group members.

Chaum and Van Heijst introduced in [9] the concept of group signature schemes. A series of improvements and enhancements followed [7], [10], [16], [17], [19], [20]. However, in these schemes the length of signatures and the size of the group's public key depend on the size of the group and thus these schemes are not suitable for large groups. The first group signature suitable for large groups is that of Camenisch and Stadler [6], where both the length of the group public key and the group signatures are independent of the group's size. The Camenisch-Stadler scheme was improved by Camenisch and Michels in [5], Ateniese and Tsudik [1] and Lee and Chang [14].

Our group signature scheme, suitable for large groups, is based on Okamoto-Shiraishi's assumption [18] (the difficulty of computation of approximate *e*-th roots modulo a composite number) and is more efficient than the previous ones.

2. The Model of Group Signature Scheme

Group signature schemes are defined as follows (see [8] for more details).

Definition 1 *A group signature scheme is a digital signature scheme comprising the following:*

1. ***Setup**: On input of a security parameter 1^l this probabilistic algorithm outputs the initial group public key P and the secret key S 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 a group member and a user whereby a group signature on a user supplied message is computed by the group member.*

4. ***Verify**: An algorithm for establishing the validity of a group signature given a group public key and a signed message.*

5. ***Open**: An algorithm that, given a signed message and a group secret key, determines the identity of the signer.*

A secure group signature scheme must satisfy the following properties:

1. Correctness: Signature produced by a group member using Sign must be accepted by Verify.

2. Anonymity: Given a signature, identifying the actual signer is computationally difficult for everyone but the group manager.

3. Unlinkability: Deciding whether two different signatures have been computed by the same group member is computationally hard.

4. No framing: Even if the group manager and some of the group members collude, they cannot sign on behalf of non-involved group members.

5. Traceability: The group manager can always establish the identity of the member who issued a valid signature.

6. Coalition-resistance: A colluding subset of group members cannot generate a valid group signature that cannot be traced.

Definition 2 *The efficiency of a group signature scheme is typically based on the size of the group public key P, the length of signature and the efficiency of the algorithms **Sign**, **Verify**, **Setup**, **Join** and **Open**.*

3. Preliminaries

This Section reviews some cryptographic assumptions and introduces the building blocks necessary for the subsequent design of our group signature scheme. The Strong RSA Assumption was independently introduced by Baric and Pfitzman [2] and Fujisaki and Okamoto [11].

Definition 3 (Strong RSA Problem) *Let n=pq be an RSA-like modulus and let G be a cyclic subgroup of Z_n^* of order l_G. Given n and $z \in G$, the Strong RSA Problem consists of finding $u \in G$ and $e \in Z_{>1}$ satisfying $z \equiv u^e \pmod n$.*

Assumption 1 (Strong RSA Assumption) *There exists a probabilistic polynomial time algorithm K which on input 1^{l_G} outputs a pair (n, z) such that for all probabilistic polynomial-time algorithms P the probability that P can solve the Strong RSA Problem is negligible.*

Assumption 2 (Decisional Diffie-Hellman Assumption) *Let n=pq be an RSA-like modulus and let α be a quadratic residue modulo n that has a large order in Z_n^*. Let $G=<\alpha>$. Given as input a triplet $(\alpha^a, \alpha^b, \alpha^c)$ in G^3, it is hard to decide whether $(\alpha^a, \alpha^b, \alpha^c)$ is a Diffie-Hellman triplet $(\alpha^a, \alpha^b, \alpha^{ab})$ or a random triplet.*

For this assumption see [4] for more details.

Assumption 3 (Okamoto-Shiraishi Assumption) *Let e be an integer, $e \geq 4$. Given as inputs an element n=pq be an RSA-like modulus and an element $C \in Z_n^*$, it is hard to find two integers X and δ such that $X^e \equiv C + \delta \pmod n$ and $\delta \in [a,b]$, where a and b are two integers satisfying $0 \leq a < b < n^{2/3}$.*

In [18] the security of the Okamoto-Shiraishi signature scheme is based on this assumption and when the exponent $e \geq 4$, the Okamoto-Shiraishi scheme is considered as robust.

Next, we present some well- studied techniques for proving knowledge of discrete logarithms. A signature of knowledge is a construct that uniquely corresponds to a given message *m* that cannot be obtained without the help of a party that knows a secret such as that of the discrete logarithm of a given $y \in G$ to the base g ($G=<g>$). We note $x=\text{Dlog}_g y$. Let *H* be a collision resistant hash function $H:\{0,1\}^* \rightarrow \{0,1\}^k$ and $\varepsilon > 1$ a security parameter. A proof of knowledge is a way for one person to convince another person that he/she knows some fact without actually revealing that fact. A signature of knowledge is used for both the purpose of signing a message and of proving knowledge of a secret.

The next building block is an adaptation of protocols for proving the knowledge of a discrete logarithm [11], and their construction, to prove knowledge, is based on the Schnorr signature scheme [21].

Definition 4 *A pair $(c,s) \in \{0,1\}^k \times \{0,1\}^{\varepsilon(l_G+k)+1}$ satisfying $c=H(m \| y \| g \| g^s y^c)$ is a signature of knowledge of the discrete logarithm of $y=g^x$ w.r.t. base g, on a message $m \in \{0,1\}^*$.*

An entity knowing the secret $x=\text{Dlog}_g y$ is able to compute the signature by choosing a random $t \in \{0,1\}^{\varepsilon(l_G+k)}$ and then computing *c* and *s* as $c=H(m \| y \| g \| g^t)$, $s=t-cx$ (in Z).

A slight modification of the previous definition let us show the knowledge and the equality of two discrete logarithms [5] of y_1 and y_2 w.r.t. bases *g* and *h*, i.e. knowledge of an integer *x* satisfying $y_1=g^x$ and $y_2=h^x$.

Definition 5 *A pair $(c,s) \in \{0,1\}^k \times \{0,1\}^{\varepsilon(l_G+k+1)}$ satisfying $c=H(m \| y_1 \| y_2 \| g \| h \| y_1^{\,c} g^s \| y_2^{\,c} h^s)$ is a signature of knowledge of the discrete logarithm of $y_1=g^x$ w.r.t. base g and $y_2=h^x$ w.r.t. base h, on a message $m \in \{0,1\}^*$.*

An entity knowing the secret *x* is able to compute the signature, provided $x=\text{Dlog}_g y_1=\text{Dlog}_h y_2$ by choosing a random $t \in \{0,1\}^{\varepsilon(l_G+k)}$ and then computing *c* and *s* as $c=H(m \| y_1 \| y_2 \| g \| h \| g^t \| h^t)$, $s=t-cx$ (in Z). The next block is based on a proof that the secret the prover knows lies in a given interval. This is a modification of the block presented in [5].

Definition 6 *A proof of knowledge of the discrete logarithm of h w.r.t. base g and of δ w.r.t. α, which also proves that $\text{Dlog}_g h=\text{Dlog}_\alpha \delta$ and that $\text{Dlog}_g h$ is in the interval $\{2^{\ell_1}-2^{\varepsilon(\ell_2+k)+1}, \ldots, 2^{\ell_1}+2^{\varepsilon(\ell_2+k)+1}\}$ is a pair (c,s), where $c=H(m \| g \| h \| \alpha \| \delta \| g^{s-c2^{\ell_1}} h^c \| \alpha^{s-c2^{\ell_1}} \delta^c)$ and $s \in \{-(2^k-1)(2^{\ell_2}-1), \ldots, 2^{\varepsilon(\ell_2+k)}\}$.*

This signature can be computed as follows. If the signer knows an integer $x \in \{2^{\ell_1},...,2^{\ell_1}+2^{\ell_2}-1\}$ such that $h=g^x$ and $\delta=\alpha^x$, he chooses a random $t \in \{0,1\}^{\varepsilon(\ell_2+k)}$ and computes $c=H(m \| g \| h \| \alpha \| \delta \| g^t \| \alpha^t)$, $s=t-c(x-2^{\ell_1})$ (in Z).

The security of the last building block has been proven in the random oracle model [3] in [5], [11].

4. Our Group Signature Scheme

In this Section we construct a new group signature scheme which is based on the Okamoto-Shiraishi assumption [18].

4.1 Setup

The group manager chooses random primes p', q' and computes p=2p'+1 and q=2q'+1. Then, the group manager computes $n=pq$. Let ℓ_n denote the bit-length of n. He chooses a public exponent e>4 such that e is relatively prime to $\varphi(n)$. The group manager selects g an element of Z_n^* of order n. Let G=<g>. The group manager selects an element $h \in G$ whose discrete logarithm to the base g must not be known. He chooses an element $C \in Z_n^*$. The group manager chooses a secret value $x \in Z_n^*$ and computes $y=g^x \pmod n$. Finally, a collision-resistant hash function $H: \{0,1\}^* \rightarrow \{0,1\}^k$ and security parameters $\varepsilon>1, \ell_1, \ell_2$ are set. An example for choosing the parameters $\varepsilon, k, \ell_G, \ell_n, \ell_1, \ell_2$ is given in Section 5. The public key is $P=(n, e, g, y, h, C, \ell_n, \varepsilon, \ell_1, \ell_2)$ and the secret key is $S=(p',q',x)$. In practice, components of P must be verifiable to prevent framing attacks (e.g. see [13]).

4.2 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 our group signature scheme consists of a pair of integers (X,δ) satisfying $X^e \equiv C+\delta \pmod n$ and $\delta \in [2^{\ell_1}, 2^{\ell_1}+2^{\ell_2}-1]$. 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 $x_i \in [2^{\ell_1}, 2^{\ell_1}+2^{\ell_2}-1]$ and computes $ID_i=g^{xi} \pmod n$.
2. The user U_i must prove to the group manager that he knows $Dlog_g ID_i$ and that this value is in the interval $(2^{\ell_1}-2^{\varepsilon(\ell_2+k)+1}, 2^{\ell_1}+2^{\varepsilon(\ell_2+k)+1})$.
3. 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.
4. 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.
5. 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.

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.

4.3 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 as follows:

1. Choose an integer $w \in_R \{0,1\}^{\ell_2}$ and compute

   $A=A_i h^w \pmod n$, $B=g^w \pmod n$, $D=g^{x_i} y^w \pmod n$.

2. Choose $r_1 \in_R \{0,1\}^{\varepsilon(\ell_{2+k})}$, $r_2 \in_R \{0,1\}^{\varepsilon(\ell_{G+}\ell_{1+k})}$, $r_3 \in_R \{0,1\}^{\varepsilon(\ell_{G+k})}$, $r_4 \in_R \{0,1\}^{\varepsilon(\ell_{2+k})}$, $r_5 \in_R \{0,1\}^{\varepsilon(\ell_{2+k})}$ and compute

   $$d_1=B^{r1}/g^{r2} \pmod n$$
   $$d_2=g^{x_i 2} D^{r4}/y^{r5} \pmod n$$
   $$d_3=g^{r3} \pmod n$$
   $$d_4=g^{r1} y^{r3} \pmod n.$$

3. Compute

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

4. Compute $s_1=r_1-c(x_i-2^{\ell_1})$, $s_2=r_2-cx_i w$, $s_3=r_3-cw$, $s_4=r_4+x_i+c2^{\ell_1}$, $s_5=r_5+x_i w+c2^{\ell_1}$ (in Z).

5. Send the group signature $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ to the verifier.

4.4 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 $c'=H(m \| g \| h \| y \| A \| B \| D \| B^{s1-c2^{\ell_1}}/g^{s2} \pmod n \| D^{s4-c2^{\ell_1}}/y^{s5-c2^{\ell_1}} \pmod n \| B^c g^{s3} \pmod n \| D^c g^{s1-c2^{\ell_1}} y^{s3} \pmod n)$.

2. 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

   $s_1 \in \{-2^{\ell_{2+k}},\ldots,2^{\varepsilon(\ell_{2+k})}\}$, $s_2 \in \{2^{\ell_{G+}\ell_{1+k}},\ldots,2^{\varepsilon(\ell_{G+}\ell_{1+k})}\}$,

   $s_3 \in \{2^{\ell_{G+k}},\ldots,2^{\varepsilon(\ell_{G+k})}\}$, $s_4 \in \{2^{\ell_{2+k}},\ldots,2^{\varepsilon(\ell_{2+k})}\}$, $s_5 \in \{-2^{\ell_{2+k}},\ldots,2^{\varepsilon(\ell_{2+k})}\}$.

4.5 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:

1. Recover ID_i (the identity of the user U_i) as $ID_i=D/B^x \pmod n$.

2. Prove that $Dlog_g v=Dlog_B(D/ID_i \bmod n)$ (see Definition 5).

5. Security and Efficiency of Our Scheme

The security of our group signature scheme is based on the difficulty of computation of approximate e-th roots modulo a composite number (see Assumption 3). We have to show that our group signature scheme satisfies all the properties listed in Section 2.

Correctness: By inspection.

Anonymity: Given a group signature, to identify the actual signer is computationally hard to do for everyone but the group manager. As the interactive protocol underlying the group signature is statistically zero-knowledge, no information is statistically revealed by $(c, s_1, s_2, s_3, s_4, s_5)$ in the random oracle model. Therefore, anonymity can only be revoked from (A, B, D). Since no one knows which pair (A_i, x_i) corresponds to which group member, anonymity is guaranteed.

Unlinkability: To decide whether two different group signatures have been computed by the same group member is computationally hard. Let $(c, s_1, s_2, s_3, s_4, s_5, A, B, D)$ and $(c', s'_1, s'_2, s'_3, s'_4, s'_5, A', B', D')$ be these signatures. The problem of linking two signatures reduces to decide whether (A, B, D) and (A',B',D') are linked or not. This can be done by looking either A_i or x_i as common to the two triples. This is impossible under Decisional Diffie-Hellman Assumption.

No framing: Neither can a group member or the group manager sign on behalf of other group members. The group manager does not get any information about a group memeber's secret x_i apart from $ID_i=g^{x_i}$. Thus, the value x_i is computationally hidden from the group manager. Hence, not even can the group manager sign on behalf of a group member U_i since computing of discrete logarithms is assumed to be nonfeasible.

Traceability: The group manager is able to open any valid group signature and provably identify the actual signer. The group manager has to issue $Dlog_g y=Dlog_B(D/ID_i \ mod \ n)$ as evidence that he decrypted the pair (B,D) correctly and thus the user U_i, who issued the group signature, can be identified.

Coalition resistance: A group certificate (A_i,x_i) can be generated only by the group manager via the Join protocol. Hence, our group signature scheme is coalition resistant.

Our group signature scheme is slightly more efficient than Camenisch-Michels's scheme [5] and relies on different security assumptions. We propose to use our group signature scheme with the following parameters: l_n=1200, e=5, ε=5/4, k=160, l_1=350, l_2=240.

6. Conclusion

In this paper we proposed a new group signature scheme suitable for large groups, i.e. the length of the group's public key and of signatures does not depend on the size of the group. The security of our group signature scheme is based on the difficulty of computation of approximate e-th roots modulo a composite number.

Acknowledgment

We gratefully acknowledge the discussions with Ivan Damgaard from DAIMI Aarhus, Denmark.

REFERENCES

1. ATENIESE, G. and TSUDIK, G., **Group Signature à la carte** , 10 th Annual ACM-SIAM Symposium on Discrete Algorithms SODA'99, 1999.
2. BARIC, N. and PFITZMANN, B., **Collision-free Accumulators and Fail-stop Signature Schemes Without Trees**, Advances in Cryptology-EUROCRYPT '97, Lecture Notes in Computer Science, Vol. 1233, SPRINGER-VERLAG, 1997, pp. 480-494.
3. BELLARE, M. and ROGAWAY, P., **Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols**, 1st ACM Conference on Computer and Communication Security, ACM PRESS, 1993, pp.62-73.
4. BONEH, D., **The Decision Diffie-Hellman Problem**, Algorithmic Number Theory (ANTS-III), Lecture Notes in Computer Science, Vol. 1423, SPRINGER-VERLAG, 1998, pp. 48-63.
5. CAMENISCH, J. and MICHELS, M., **A Group Signature Scheme With Improved Efficiency**, Advances in Cryptology-ASIACRYPT'98, Lecture Notes in Computer Science, Vol. 1514, SPRINGER-VERLAG, 1998, pp. 160-174.
6. CAMENISCH, J. and STADLER, M., **Efficient Group Signature Schemes for Large Groups**, Advances in Cryptology-CRYPTO '97, Lecture Notes in Computer Science, Vol. 1296, SPRINGER-VERLAG, 1997, pp. 410-424.
7. CAMENISCH, J., **Efficient and Generalized Group Signatures**, Advances in Cryptology-EUROCRYPT'97, Lecture Notes in Computer Science, Vol. 1233, SPRINGER-VERLAG, 1997, pp. 465-479.
8. CAMENISCH, J., **Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem**, Ph. D Thesis, Vol. 2, ETH Series in Information Security on Cryptography, HARTUNG-GORRE VERLAG, 1998.
9. CHAUM, D. and VAN HEYST, E., **Group Signatures**, Advances in Cryptology-EUROCRYPT '91, Lecture Notes in Computer Science, Vol. 547, SPRINGER-VERLAG, 1991, pp. 257-265.
10. CHEN, L. and PEDERSEN, T., **New Group Signature Schemes**, Advances in Cryptology-EUROCRYPT '94, Lecture Notes in Computer Science, Vol. 950, SPRINGER-VERLAG, 1995, pp. 171-181.
11. FUJISAKI, E. and OKAMOTO, T., **Statistical Zero Knowledge Protocols To Prove Modular Polynomial Relations**, Advances in Cryptology-CRYPTO '97, Lecture Notes in Computer Science,Vol. 1297, SPRINGER-VERLAG, 1997, pp. 16-30.
12. FUJISAKI, E. and OKAMOTO, T., **A Practical and Provably Secure Scheme for Publicly Verifiable Secret Sharing and Its Applications**, Advances in Cryptology-EUROCRYPT '98, Lecture Notes in Computer Science, Vol. 1403, SPRINGER-VERLAG, 1998, pp. 32-46.
13. GENARO, R., MICCIANCIO, D. and RABIN, T., **An Efficient Non-interactive Statistical Zero-knowledge Proof System for Quasi-safe Prime Products**, Proceedings of the 5 th ACM Conference on Computer and Communications Security, 1998.
14. LEE, W. and CHANG, C., **Efficient Group Signature Scheme Based On the Discrete Logarithm**, IEE Proceedings Comput. Digit. Tech. 145, No. 1, 1998, pp. 15-18.
15. LYSYANSKAYA, A. and RAMZAN, Z., **Group Blind Signature: A Scalable Solution to Electronic Cash**, Financial Cryptography (FC '98), Lecture Notes in Computer Science, Vol. 1465, SPRINGER-VERLAG, 1998, pp. 184-197.
16. KIM, S., PARK, S. and WON, D., **Convertible Group Signatures,** Advances in Cryptology- ASIACRYPT '96, Lecture Notes in Computer Science, Vol. 1163, SPRINGER-VERLAG, 1996, pp. 311-321.
17. KIM, S., PARK, S. and WON, D., **ID-based Group Signature Schemes**, ELECTRONICS LETTERS, 1997, pp. 1616-1617.
18. OKAMOTO, T. and SHIRAISHI, A., **A Fast Signature Scheme Based On Quadratic Inequalities**, Proceedings of IEEE Symposium on Security and Privacy, 1985, pp. 123-132.

19. PETERSEN, H., **How To Convert Any Digital Signature Scheme Into A Group Signature Scheme**, Security Protocols Workshop, Paris, 1997.

20. POPESCU, C., **A Group Signature Scheme Based On the Discrete Logarithm Problem**, to appear in Studia Universitatis Babes-Bolyai, Series Computer-Science, Cluj -Napoca.

21. SCHNORR, C. P., **Efficient Signature Generation for Smart Cards**, JOURNAL OF CRYPTOLOGY, Vol.4, No. 3, 1991, pp. 239-252.