

## GENERALIZED GROUP SIGNATURE BASED ON AN RSA-VARIANT

EDDIE SHAHRIL ISMAIL<sup>1</sup>, YAHYA ABU HASAN<sup>2</sup>, & HOW GUAN AUN<sup>3</sup>

**Abstract.** A standard group signature scheme allows a group member (single user) to sign messages anonymously on behalf of the group but in case of a dispute, the identity of an actual signer can be revealed by a designated entity. This paper generalizes the concept by allowing not a single user but a number of users (forming a group) to sign messages anonymously on behalf of the group. In case of a legal dispute, a designated entity can trace or determine which group has signed a given message. We build a generalized group signature scheme based on Multiple RSA—a variant of celebrated RSA. Multiple RSA is a public key cryptosystem, which permits a number of users to decrypt a ciphertext jointly. The realization of our generalized group signature scheme is mainly due to the key generation algorithm in Multiple RSA. We use this key generator to initialize the signing procedure. Thus our scheme provides an extra security since the users have to show two certificates to a designated entity before allowing them from signing messages.

*Keywords:* Digital signature, group signature, RSA, multiple RSA, decisional Diffie-Hellman assumption, zero-knowledge proofs

**Abstrak.** Skema tandatangan kumpulan piawai membenarkan seorang ahli kumpulan menandatangani mesej bagi pihak kumpulannya tanpa diketahui identiti ahli itu tetapi jika berlaku kekalutan, identiti ahli itu boleh dibongkar oleh entiti tertentu. Artikel ini memperluaskan konsep ini dengan sebilangan ahli (membentuk kumpulan) boleh menandatangani mesej bagi pihak kumpulan tanpa diketahui identiti sebilangan ahli itu. Untuk kes kekalutan, entiti tertentu dapat menjejak kumpulan yang menandatangani mesej itu. Kami membina skema perluasan tandatangan kumpulan ini berdasarkan kepada kriptosistem Multi-RSA suatu variasi daripada RSA. Multi-RSA adalah kriptosistem kunci awam untuk sebilangan pelanggan menyahsilit mesej secara bersama. Skema ini direalisasikan khusus daripada algoritma penjaan kunci dalam Multi-RSA. Kami menggunakan penjana kunci ini untuk memulakan prosedur tandatangan. Skema kami menyediakan keselamatan tambahan kerana ahli-ahli kumpulan mesti menunjukkan dua sijil kelayakan kepada entiti sebelum dibenarkan menandatangani mesej-mesej.

*Kata kunci:* Tandatangan digital, tandatangan kumpulan, RSA, multi RSA, penentuan andaian Diffie-Hellman, pembuktian pengetahuan-sifar

<sup>1</sup> No. 9, Rumah KTMB, Simpang 5, 34200 Parit Buntar, Perak Darul Ridzuan. Tel: 05-7160821/012-5539407. e-mail: eshahril@hotmail.com/esbi@pkrisc.cc.ukm.my

<sup>2</sup> Pusat Pengajian Sains Matematik, Universiti Sains Malaysia, 11800 USM Minden Penang. Tel: 04-6577888 ext. 3966. Fax: 04-6570910. e-mail: ahyahya@cs.usm.my

<sup>3</sup> Pusat Pengajian Sains Matematik, Universiti Sains Malaysia, 11800 USM Minden, Penang. Tel: 04-6577888 ext. 2428. Fax: 04-6570910. e-mail: gahow@cs.usm.my

## 1.0 INTRODUCTION

Group signature, introduced by David Chaum and Van Heyst in 1991 was another variant concept from the traditional digital signature. In contrast to ordinary signatures, group signature provides anonymity to the signer. Thus a verifier can only tell that the message has been signed without being able to reveal the identity of a signer except for a designated entity, who can revoke the anonymity if there is a need to do so (such as in a legal dispute). For an in-depth discussion on group signature, please consult Camenisch [1].

In this work, we introduce a generalized group signature scheme based on Multiple RSA cryptosystem-a variant from the celebrated RSA. See Aun et al., [2] for details. An ordinary group signature allows a single signer to sign messages anonymously on behalf of the group but in our scheme, a group (consists of a number of users) is allowed to do so. However, if a legal dispute happens, our designated entity can identify which group has issued the signature.

The first efficient and generalized group signature scheme can be found in Camenisch [1]. Basically, our scheme is another contribution and realization to the theory proposed in that paper. Our scheme however, relies on the idea of Ateniese et al., [3]. The proposed scheme was significantly more efficient and secure than the state of the art.

Now we briefly review the Multiple RSA cryptosystem. We will focus only on the key generation, where a designated entity generates and distributes the public and private keys to all users in the system. These keys will then be used in the initializing phase of our generalized group signature scheme.

## 2.0 MULTIPLE RSA CRYPTOSYSTEM

Let the system contains  $k$  users and  $P = \{P_1, \dots, P_k\}$  denote a set of all  $k$  members and GM is a designated entity, called the group manager. Let  $\Gamma = \{G_i : G_i \subseteq P, |G_i| = s\}$  be a set of all authorized groups and each group must consist of  $s$  users and we denote users in each group as  $\{P_{i_1}, \dots, P_{i_s}\} = G_i$ . GM chooses two large primes  $p$  and  $q$  (for security, 512-bits primes are required as the standard size for the real world applications) such that  $(s, \phi(n)) = 1$  where  $\phi(n) = (p-1)(q-1)$  is a Euler function and  $n = pq$ . GM next computes a secret  $r \in Z_{\phi(n)}^*$  from

$$rs \equiv 1 \pmod{\phi(n)}. \quad (2.01)$$

Then GM chooses a set of public keys  $E = \{e_1, \dots, e_k\}$  satisfying  $(e_i, \phi(n)) = 1$  and  $(e_i, e_j) \neq 1$  for two different public keys. Then a set  $D = \{d_1, \dots, d_k\}$  containing  $k$  private keys is chosen from

$$e_i(d_i + d) \equiv r \pmod{\phi(n)} \quad (2.02)$$

where  $d$  is a GM's secret key. GM now distributes  $E$  and  $D$  to users in the system. Obviously, GM knows everyone's key thus we assume that he is trustworthy and

credible. Now we give a model of standard group signature scheme. This model is also valid for generalized group signature scheme.

### 3.0 THE MODEL

**Definition 3.1.** A group signature scheme is a digital signature scheme containing the following five procedures:

- (1) **SETUP:** An algorithm for generating the initial group public key and the secret key for the group manager.
- (2) **JOIN:** A protocol between the group manager and a user that results in the user becoming a new group member. The user's output is a membership certificate and a membership secret.
- (3) **SIGN:** An algorithm that on input a group public, a membership certificate, a membership secret, and a message  $m$ , outputs group signature of  $m$ .
- (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 message, a valid group signature, a group public key and a group manager's secret key, determines the identity of the signer.

A group signature scheme is secure if the following conditions are satisfied:

- (a) **Correctness:** Signatures produced by a group member using SIGN must be accepted by VERIFY.
- (b) **Unforgeability:** Only group members are able to sign on behalf of the group.
- (c) **Anonymity:** Given a valid signature of some messages, identifying the actual signer is computationally difficult for everyone but the group manager.
- (d) **Unlinkability:** Deciding whether two different valid signatures were computed by the same group member is computationally challenging.
- (e) **Exculpability:** Neither a group member nor the group manager can sign on behalf of other group members.
- (f) **Traceability:** The group manager is always able to open a valid signature and identify the actual signer.

We observe that some group signature schemes excluded the JOIN protocol. This is because such schemes were constructed under some known cryptosystems. As in our case, we constructed a group signature scheme under the Multiple RSA system but without loss of generality, the JOIN protocol is included. Thus, users in a qualified group seem to have 'two certificates' before getting a permission in signing procedure. This gives an extra security to the scheme. If a group fails to run the JOIN protocol, then they are prohibited any message from signing in our group signature scheme.

## 4.0 CRYPTOGRAPHIC ASSUMPTIONS AND BUILDING BLOCKS

Before explaining in detail on our group signature scheme, we will review a cryptographic assumption and introduce the building blocks. We use the following cryptographic assumption in order to achieve some standard security by relying on some intractable facts. The introduced building blocks are the backbone of our group signature scheme.

We design the scheme based on these building blocks, which is an adaptation from Schnorr's signature scheme (Camenisch [1]). This scheme can be used to prove some knowledge of facts to the other party without revealing the secret information.

### 4.1 Number-Theoretic Assumptions

For more information and thorough discussion on Decisional Diffie-Hellman Problem and Decisional Diffie-Hellman Assumption, please refer to Boneh [4].

Listed below are some notations and we will use them throughout the paper:

- (a)  $a \in_R Z$  denotes  $a$  is chosen at random in  $Z$ .
- (b)  $a || b$  denotes the concatenation of two binary strings  $a$  and  $b$ .

**Definition 4.1. (Decisional Diffie-Hellman Problem).** Let  $G = \langle g \rangle$  be a cyclic group generated by  $g$  of order  $\#G$ . Given  $g, g^x, g^y$  and  $g^z \in G$ , the Decisional Diffie-Hellman Problem consists of deciding whether the elements  $g^{xy}$  and  $g^z$  are equal.

**Assumption 4.1. (Decisional Diffie-Hellman Assumption).** There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between the distributions  $D = (g, g^x, g^y, g^z)$  and  $R = (g, g^x, g^y, g^{xy})$  with  $x, y \in_R Z_{\#G}$ .

### 4.2 Signatures of Knowledge

Zero-knowledge proofs of knowledge allow a prover to demonstrate the knowledge of a secret without revealing any secret information. Showing the knowledge of the discrete logarithm of  $y = g^x$  can be shown easily. We call the resulting construction as signature of knowledge due to Camenisch et al. [5]. Before we go further, we assume that a security parameter  $\varepsilon > 1$  and let  $H$  be a collision-resistant hash function

$H : \{0,1\}^* \rightarrow \{0,1\}^l$  where  $l \approx 160$  and details on hash function and its importance can be found in Schneier [6].

**Definition 4.2.** Let  $y, g \in G$ . A pair  $(c, s) \in \{0, 1\}^k \times \pm\{0, 1\}^{\varepsilon(l_G+k)+1}$  verifying  $c = H(y \parallel g \parallel g^s y^c \parallel m)$  is a signature of knowledge of the discrete logarithm of  $y = g^x$  with respect to the base  $g$ , on a message  $m \in \{0, 1\}^*$ .

The party in possession of the secret  $x = \log_g y$  is able to compute the signature by choosing a random  $t \in \pm\{0, 1\}^{\varepsilon(l_G+k)}$  and then computing  $c$  and  $s$  as:

$$c = H(y \parallel g \parallel g^t \parallel m) \text{ and } s = t - cx \text{ (in } \mathbb{Z}\text{)}.$$

From the above definition, we can modify to show the knowledge and equality of  $h$  discrete logarithms of, say  $y_i, i = 1, \dots, h$ , with respect to the bases  $g_i, i = 1, \dots, h$ , i.e., knowledge of an integer  $x$  satisfying  $y_i = g_i^x, i = 1, \dots, h$ .

**Definition 4.3.** Let  $y_i, g_i \in G$ , for  $i = 1, \dots, h$ . A pair  $(c, s) \in \{0, 1\}^k \times \pm\{0, 1\}^{\varepsilon(l_G+k)+1}$  verifying  $c = H(y_1 \parallel \dots \parallel y_h \parallel g_1 \parallel \dots \parallel g_h \parallel g_1^s y_1^c \parallel \dots \parallel g_h^s y_h^c \parallel m)$  is a signature of knowledge of the discrete logarithm of  $y_i = g_i^x$  with respect to the bases  $g_i, i = 1, \dots, h$ , on a message  $m \in \{0, 1\}^*$ .

The party in possession of the secret  $x$  is able to compute the signature, provided that  $x = \log_{g_i} y_i, i = 1, \dots, h$ , by choosing a random  $t \in \pm\{0, 1\}^{\varepsilon(l_G+k)+1}$  and then computing  $c$  and  $s$  as:

$$c = H(y_1 \parallel \dots \parallel y_h \parallel g_1 \parallel \dots \parallel g_h \parallel g_1^t \parallel \dots \parallel g_h^t \parallel m) \text{ and } s = t - cx \text{ (in } \mathbb{Z}\text{)}.$$

The above efficient proofs (or signatures) of knowledge of discrete logarithms and its generalization can be used for realizing an efficient group signature scheme.

## 5.0 THE GENERALIZED GROUP SIGNATURE SCHEME

Let say a group  $G_i$  wishes to sign a message anonymously on behalf of  $P$ . Before that, all users in  $G_i$  must run the JOIN protocol secretly with the GM in order to get a membership/group certificate. This membership certificate acts as permission for users to run the next procedures.

The initial phase involves the GM setting the group public and his secret keys.

*SETUP:*

- (1) Select all parameters and generate all keys as in Multiple RSA.
- (2) Choose two random elements  $T, L \in_R QR(n)$  and set  $S \equiv T^d \pmod{n}$ .
- (3) The group public key is  $PK = (n, S, T, L)$ .
- (4) The secret key (known only to GM) is  $SK = (p, q, d)$ .

Here  $QR(n)$  is a subgroup of quadratic residues modulo  $n$ . Note that, the public key  $PK$  does not consists of user's public key, since our verification algorithm does not require that. Those keys will be used only in signing algorithm. Note also, our group public key does not depend on the size of the group. Clearly, this is an advantage.

*JOIN:*

- (1) Each  $P_{i_j} \in G_i$  sends GM a pair  $(e_{i_j}, d_{i_j})$  as a proof of member in Multiple RSA.
- (2) If the pair matched (to GM's secret list/record), he sends  $P_{i_j}$  a random element  $x_{i_j} \in_R QR(n)$ .
- (3) Then all users in that group agree on a common secret exponent,  $\omega_i \in_R Z_n^*$ .
- (4) Each user then computes his partial membership secret and sends  $y_{i_j} \equiv x_{i_j}^{\omega_i} \pmod{n}$  to GM and proves him that he (user) knows the discrete logarithm of  $y_{i_j}$  with respect to the base  $x_{i_j}$ .
- (5) GM checks that  $y_{i_j} \in QR(n)$  for all  $j = 1, \dots, s$ . If this is the case and the proof (4) was correct, together with the proof of knowledge of equality of  $s$  discrete logarithms of  $y_{i_j}$  with respect to the base  $x_{i_j}$ , for all  $j = 1, \dots, s$  then GM computes two integers  $u_i$  and  $v_i$  such that,  $u_i v_i \equiv 1 \pmod{\phi(n)}$ .
- (6) GM next computes  $A_i \equiv \left( \prod_{j=1}^s y_{i_j} \right)^{v_i} \pmod{n}$ . Finally, GM sends group  $G_i$  the membership/group certificate  $[A_i, u_i]$ .
- (7) Users in  $G_i$  get together and verify that  $\prod_{j=1}^s y_{i_j} \equiv A_i^{u_i} \pmod{n}$ .

*Remark 5.1.* GM next stores  $\{[A_i, u_i], ID\}$  in the membership table. The  $ID$  should consists, for example any type of identification of  $G_i$ . It seems infeasible to construct

such a triple  $\left( A_i, \gamma = \prod_{j=1}^s y_{i_j}, \omega_i \right)$  without the help of the GM. If  $\gamma$  is computed correctly,

then it is infeasible to obtain  $A_i$  because nobody knows  $v_i$ . Someone who knows  $u_i$  still learns nothing because the factorization of  $n$  is unknown. The common secret exponent,  $\omega_i$ , is also difficult to compute since computing the discrete logarithm problem is infeasible. Thus we conclude that, only GM can generate the membership/group certificate.

Armed with a membership/group certificate, users in  $G_i$  can now generate anonymous and unlinkable group signatures on a message  $m \in \{0,1\}^*$ .

*SIGN:*

- (1) Each user  $P_{i_j}$  computes:

$$F_{1i_j} \equiv A_i^{e_{ij} d_{ij}} \pmod{n}, F_{2i_j} \equiv A_i^{e_{ij}} \pmod{n} \text{ and } F_{3i_j} \equiv F_{1i_j} F_{2i_j} \pmod{n}.$$

- (2) Next, all users agree on a common secret,  $\lambda_i \in_R Z_n^*$ .  
 (3) Then they compute:

$$T_{1i} \equiv \left( \prod_{j=1}^s F_{1i_j} \right) L^{\lambda_i} \pmod{n}, T_{2i} \equiv \left( \prod_{j=1}^s F_{2i_j} \right) T^{\lambda_i} \pmod{n},$$

$$T_{3i} \equiv (LS)^{\lambda_i} \pmod{n} \text{ and } T_{4i} \equiv \left( \prod_{j=1}^s F_{3i_j} \right)^{\omega_i \lambda_i} \pmod{n}.$$

- (4) They next cooperate and choose three random elements  $t_{1i}, t_{2i}, t_{3i} \in_R Z_n^*$  and compute:

- (a)  $h_{1i} \equiv (T_{1i} T_{2i})^{t_{1i}} / (LT)^{t_{2i}} \pmod{n}$ ,  $h_{2i} \equiv (LS)^{\lambda_i t_{1i} - t_{2i}} \pmod{n}$  and  
 $h_{3i} \equiv (LS)^{t_{3i}} \pmod{n}$ ;  
 (b)  $c = H(S \| T \| L \| T_{1i} \| T_{2i} \| T_{3i} \| T_{4i} \| h_{1i} \| h_{2i} \| h_{3i} \| m)$ ;  
 (c)  $s_{1i} = t_{1i} - c \omega_i \lambda_i$ ,  $s_{2i} = t_{2i} - c \omega_i \lambda_i^2$  and  $s_{3i} = t_{3i} - c \lambda_i$  (all in  $Z$ );

- (5) Output  $(c, s_{1i}, s_{2i}, s_{3i}, T_{1i}, T_{2i}, T_{3i}, T_{4i})$  as a signature on  $m$ .

Note that, GM does not involve in the signing procedure and he cannot even sign the message on group's behalf.

A verifier can check the validity of a signature of message  $m$  as follows:

*VERIFY:*

- (1) Compute:  $c' = H \left( S \| T \| L \| T_{1i} \| T_{2i} \| T_{3i} \| T_{4i} \| \right.$   
 $\left. (T_{1i} T_{2i})^{s_{1i}} T_{4i}^c / (LT)^{s_{2i}} \| T_{3i}^{s_{1i}} / (LS)^{s_{2i}} \| T_{3i}^c (LS)^{s_{3i}} \| m \right)$   
 (2) Accept the signature if and only if  $c = c'$  and  $s_{ji} \in [1, \dots, n]$  for  $j = 1, 2, 3$ .

In the event of legal dispute, GM executes the following procedure to identify which group has produced the signature.

*OPEN:*

- (1) Check the signature's validity via the VERIFY procedure.
- (2) Recover  $A_i$  as  $A_i \equiv (T_{1i} T_{2i}^d / T_{3i}) \pmod{n}$ .
- (3) Prove that  $\log_T S \equiv \log_{T_{2i}} (A_i T_{3i} / T_{1i}) \pmod{n}$ .

## 6.0 SECURITY ANALYSIS

The presented scheme allows any group in  $\Gamma$  to sign messages anonymously on behalf of  $P$ . In the event of legal dispute, GM can identify the signature's originator, namely, the group that has produced the signed messages. Once the scheme or system has been set up, the combination of  $s$  is fixed. However, our scheme permits any new user to join the group signature scheme and it is not necessary to modify our group public key. Let say, the  $(k+1)$ -th user  $P_{k+1}$  wishes to join the scheme. All he has to do is to get a key pair  $(e_{k+1}, d_{k+1})$  from the GM. He then with his intended group can now sign messages anonymously on behalf of  $P$ .

Note that, there is no relationship between the length of both  $PK$  and the signature and the size  $P$ . Thus the running time of the verification and signing algorithms are independent to the number of group members.

Now we show that our scheme is secure since all requirements in Definition 3.1 are satisfied.

**Correctness:** By inspection.

**Unforgeability:** Only valid groups are able to sign messages on behalf of  $P$ . Besides the possession of group certificate  $[A_i, u_i]$ , users in  $G_i$  also have to be a member of the system. Without these two qualifications, the group  $G_i$  is not permitted to involve in the signing procedure.

**Anonymity:** Everyone sees a valid group signature  $(c, s_{1i}, s_{2i}, s_{3i}, T_{1i}, T_{2i}, T_{3i}, T_{4i})$  but cannot later determines which group has issued the signature except for the GM. This is because, finding  $A_i$  is equivalent to the discrete logarithm problem [7] which is hard to solve. Deciding some groups with certificate  $[A_i, u_i]$  originated requires deciding whether the four discrete logarithms,

$$\log_L(T_{1i} / \prod_{j=1}^s F_{1j}), \log_T(T_{2i} / \prod_{j=1}^s F_{2j}), \log_{LS} T_{3i} \text{ and } \log \left( \prod_{j=1}^s F_{3ij} \right)^{\omega_i} T_{4i}$$

are equal. However, this is impossible under the Decisional Diffie-Hellman Assumption. Thus the anonymity is guaranteed.

**Unlinkability:** The problem of linking two valid signatures  $(c, s_{1i}, s_{2i}, s_{3i}, T_{1i}, T_{2i}, T_{3i}, T_{4i})$  and  $(\bar{c}, \bar{s}_{1i}, \bar{s}_{2i}, \bar{s}_{3i}, \bar{T}_{1i}, \bar{T}_{2i}, \bar{T}_{3i}, \bar{T}_{4i})$  reduces to decide whether the four discrete logarithms,

$$\log_L T_{1i}/\bar{T}_{1i}, \log_T T_{2i}/\bar{T}_{2i}, \log_{LS} T_{3i}/\bar{T}_{3i} \text{ and } \log \left( \prod_{j=1}^s F_{3ij} \right)^{\omega_i} T_{4i}/\bar{T}_{4i}$$

are equal. This is however, impossible under Decisional Diffie-Hellman Assumption.

**Exculpability:** Neither a group,  $G_j$  nor the GM can sign on behalf of other group  $G_i$ , for  $i \neq j$ . This is an immediate consequence from Remark 5.1.

**Traceability:** The GM is able to open any valid signature and identify the actual signer. Assuming that the signature is valid, a group certificate  $[A_i, u_i]$  can be recovered (only by GM). This can be done using all available information  $(T_{ji}, j = 1,2,3)$  and GM's secret key,  $d$ .

## 7.0 CONCLUSION

This paper presents a generalized group signature scheme based on Multiple RSA-a variant of RSA. The scheme allows a group to sign messages/documents anonymously on behalf of  $P$ . This is a generalization from the standard group signature, which only permits a single user to sign messages. The group signature and its generalization have many applications generally in electronic transaction environment (electronic voting), payment systems (on-line and off-line), and banking systems. The security of the scheme depends heavily on a cryptographic assumption and we believe that the scheme can be implemented successfully as a national standard.

## REFERENCES

- [1] Camenisch, J. 1998. Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem, PhD thesis, vol. 2 of *ETH Series in Information Security on Cryptography*, Hartung-Gorre Verlag, Konstanz.
- [2] Aun, H. G., Y. A. Hasan, and E. S. Ismail. 2001. Kriptosistem Multi-RSA, *Jurnal Teknologi 35(C)*, Universiti Teknologi Malaysia: 61-70.
- [3] Ateniese, G., J. Camenisch., M. Joye, and G. Tsudik. 2000. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme in *Advances in Cryptology-CRYPTO 2000, LNCS 1880*: 255-270.
- [4] Boneh, D. 1998. The Decision Diffie-Hellman Problem in *Algorithmic Number Theory (ANT-111)*, vol. 1423 of LNCS, Springer-Verlag: 48-63.

- [5] Camenisch, J., and M. Stadler. 1997. Efficient Group Signature Scheme for Large Groups. *Advances in Cryptology-CRYPTO'97*, vol. 1296 of *LNCS*, Springer-Verlag: 410-424.
- [6] Schneier, B. 1996. *Applied Cryptography*, 2<sup>nd</sup> Edition, John-Wiley.
- [7] ElGamal, T. 1985. A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithm Problem. *IEEE Trans. Info. Theory*, IT-31: 469-472.