ID-based Serial Multisignature Scheme using Bilinear Pairings*

Raju Gangishetti, M. Choudary Gorantla, Manik Lal Das,
Ashutosh Saxena and Ved P. Gulati

Institute forDevelopment and Research in Banking Technology
Castle Hills, Road #1, Masab Tank, Hyderabad 500057, AP, INDIA.

Abstract. This paper presents an ID-based serial multisignature scheme using bilinear pairings. We use Hess's ID-based signature scheme as the base scheme for our multisignature scheme. Our scheme requires a forced verification at every level to avoid the overlooking of the signatures of the predecessors. We show that the scheme is secure against existential forgery under adaptive chosen message attack in the random oracle model.

1 Introduction

Shamir [18] introduced the concept of ID-based cryptosystems where, a user's public key could be easily derived from his identity and the user's private key is generated by a trusted third party called Private Key Generator (PKG). ID-based cryptosystems are advantageous over the traditional public key cryptosystems (PKCs), as key distribution and revocation are not required. A verifier can verify a signature just by using the signer's identity.

In day-to-day life, many legal documents require signatures from more than one party e.g. contracts, decision making processes, petitions etc. To meet these requirements in the digital environment, cryptography provides a mechanism known as multisignature. A multisignature scheme provides:

- multiple signers to generate a signature for a single message
- a convincing mechanism to the verifier that each stated signer had signed the message.

A multisignature scheme is practicable when the size of the multisignature by n signers is less than the total size of n signatures in the single signature scheme, on which the multisignature scheme is based. Accordingly, the verification cost gets reduced.

Based on the nature of applications, the multisignatures have been categorized into two types: serial and parallel. In serial multisignature, a signer signs the message and sends it to the next signer for further processing; the next signer after verifying his

* This work is supported in part by the Ministry of Communications and Information Technology, Govt. of India, under the grant no. 12(35)/05-IRSD.

Gangishetti R., Choudary Gorantla M., Lal Das M., Saxena A. and P. Gulati V. (2005).
ID-based Serial Multisignature Scheme using Bilinear Pairings.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 40-47
DOI: 10.5220/0002559100400047

predecessor's signature, signs the received signed component. The message signing is considered complete when the last signer's signature is appended. In the case of parallel multisignature, the signature of each signer is carried out on the content of the message but not on the signatures of the other signers. Many financial transactions require serial multisignatures and verification at each level like maker-checker, wherein the maker, checker and approval concept is being followed in a sequence and every signer is logically forced to verify his immediate predecessor's signature. Our scheme, presented in this work requires a forced verification at every level to avoid the overlooking of the signatures of the predecessors.

In this paper, we propose an ID-based serial multisignature scheme using bilinear pairings. We use Hess's ID-based signature scheme [8] as the base scheme for our multisignature scheme. The scheme is secure against existential forgery under adaptive chosen message attack in the random oracle model assuming weak Deffie-Helman problem is hard.

The rest of the paper is organized as follows. In Section 2, we briefly review the related works. In Section 3, we describe background concepts on bilinear pairings and some related mathematical problems. In Section 4, we present our proposed serial multisignature scheme and analyze the scheme in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

In 1983 Itakura and Nakamura [10] first introduced the notion of multisignature. Since then, several schemes [2], [5], [7], [9], [11]-[13], [15]-[17] for multisignatures have been proposed. The proposal of [7] was cryptanalyzed by [9]. Ohta et al. [15] proposed a scheme to avoid the restriction on the signing order. The security analysis of the scheme [16] does not consider the key generation phase. A formal notion of security for multisignature was proposed by Micali et al.[13]. Then, Lin et al.[12] proposed a structured multisignature scheme from the Gap-Diffie-Hellman group. Recently, Boldyreva [2] proposed a generic notion of security for multisignature scheme based on bilinear pairings. In 2001 Lin et al. [11] proposed ID-based structured multisignature scheme on which successful attack was carried out by Mitchell [14]. In 2003, Chen et al.[6] proposed a multi proxy signature scheme using parallel multisignatures.

3 Background Concepts

In this section, we briefly review the basic concepts on bilinear pairings and some related mathematical problems.

3.1 Bilinear Pairings

Let G_1 be an additive cyclic group of prime order q, G_2 be a multiplicative cyclic group of the same order and P be a generator of G_1. A bilinear map e is defined as $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

Bilinear: $e(aR, bS) = e(R, S)^{ab}$ $\forall R, S \in G_1$ and $a, b \in Z_q^*$. This can be restated as $\forall R, S, T \in G_1$, $e(R + S, T) = e(R, T)e(S, T)$ and $e(R, S + T) = e(R, S)e(R, T)$.
Non-degeneracy: If P is a generator of G_1, then $e(P, P)$ is a generator of G_2. In other words $e(P, P) \neq 1$.
Computable: There exists an efficient algorithm to compute $e(R, S)$ $\forall R, S \in G_1$.

In general implementation, G_1 will be the group of points on an elliptic curve and G_2 will denote a multiplicative subgroup of a finite field. Typically, the mapping e will be derived from either the Weil or the Tate pairing on an elliptic curve over a finite field. We refer to [3] for more comprehensive description on how these groups, pairings and other parameters are defined.

3.2 Computational Problems

Now, we give some computational problems, which will form the basis of security for our schemes.
Discrete Logarithm Problem (DLP): Given two elements $R, S \in G_1$, find an integer $n \in Z_q^*$, such that $S = nR$ whenever such an integer exists.
Computational Diffie-Hellman Problem (CDHP): For any $a, b \in Z_q^*$, given $< P, aP, bP >$, compute abP.
Decisional Diffie-Hellman Problem(DDHP): For any $a, b, c \in Z_q^*$, given $< P, aP, bP, cP >$, decide whether $c \equiv ab \bmod q$.
Bilinear Diffie-Hellman Problem (BDHP): For any $a, b, c \in Z_q^*$, given $< P, aP, bP, cP >$, compute $e(P, P)^{abc}$.
Gap Diffie-Hellman Problem(GDHP): A class of problems where CDHP is hard while DDHP is easy.
Weak Diffie-Hellman Problem(WDHP): For $S \in G_1$ and for some $a \in Z_q^*$, given $< P, S, aP >$ compute aS.

4 Proposed Serial Multisignature Schemes

The entities involves in our scheme are the Private Key Generator (PKG), set of Signers $\mathbf{S}$ and the Verifier $\mathbf{V}$. The proposed serial multisignature scheme consists of four phases: Setup, Private Key Extraction, Multisignature Generation and Multisignature Verification.

4.1 Setup

PKG publishes system parameters `params`=$\{G_1, G_2, e, q, P, P_{pub}, H, h\}$, here G_1 is a cyclic additive group and G_2 is a cyclic multiplicative group with prime order q. $P(\in G_1)$ is a generator of G_1, $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map between the groups G_1 and G_2, $H : \{0,1\}^* \rightarrow G_1^*$ and $h : \{0,1\}^* \times G_2 \rightarrow Z_q^*$ where $G_1^* = G_1 \setminus \{0\}$ are the cryptographic hash functions. $P_{pub} = s_0 P$ is the public key of the PKG where $s_0 \in Z_q^*$ is master secret of the PKG.

4.2 Private Key Extraction

Let the set of n Signers with identities I={ID$_1$, ID$_2$,...,ID$_n$} $\in \{0,1\}^*$. For the signer with ID$_i$ the public key $Q_{ID_i} = H(ID_i)$ and the private key is $S_{ID_i} = s_0 Q_{ID_i}$.

4.3 Multisignature Generation

In this phase n signers with identities {ID$_1$, ID$_2$,...,ID$_n$} $\in \{0,1\}^*$ sequentially generate the multisignature and the final signer sends it to the verifier. To have a multisignature on message m, without losing the generality we present it in following stages.

Signature generation by First Signer: To sign a message m the first signer, picks a random integer $k_1 \in_R Z_q^*$ and computes

$$\begin{aligned} r_1' &= e(P,P)^{k_1} \\ r_1 &= r_1' \\ c_1 &= h(m, r_1) \\ u_1 &= c_1 S_{ID_1} + k_1 P \end{aligned}$$

The signature by the first signer is the tuple $\langle u_1, c_1 \rangle \in (G_1, Z_q^*)$ and sends to the second signer along with the message m.

Verification and Signature by intermediate (ith) Signer: The ith signer verifies the signature $\langle u_{i-1}, c_1, c_2, ..., c_{i-1} \rangle$ received from $(i-1)$th signer by computing

$$r_{i-1} = e(u_{i-1}, P) e(\sum_{j=1}^{i-1} c_j Q_{ID_j}, -P_{pub})$$

Accepts the signature if and only if $c_{i-1} = h(m, r_{i-1})$
Signature generation by ith signer as follows, picks a random integer $k_i \in_R Z_q^*$ then computes

$$\begin{aligned} r_i' &= e(P,P)^{k_i} \\ r_i &= r_{i-1} r_i' \\ c_i &= h(m, r_i) \\ u_i &= u_{i-1} + c_i S_{ID_i} + k_i P \end{aligned}$$

Then he sends the partial multisignature $\langle u_i, c_1, c_2, ..., c_i \rangle$ to the $(i+1)$th signer. One may note that ith signer cannot generate his signature without verifying the $(i-1)$th signers signatures, because for computing r_i the ith signer has to extract r_{i-1} from the signature $\langle u_{i-1}, c_1, c_2, ..., c_{i-1} \rangle$ received. This confirms the correctness of the predecessors signature.

Verification and Signature by the final(nth) Signer: The nth signer verifies the signature $\langle u_{n-1}, c_1, c_2, ..., c_{n-1}\rangle$ received from $(n-1)$th signer by computing

$$r_{n-1} = e(u_{n-1}, P)e(\sum_{j=1}^{n-1} c_j Q_{ID_j}, -P_{pub})$$

Accepts the signature if and only if $c_{n-1} = h(m, r_{n-1})$
Signature generation by nth signer as follows, picks a random integer $k_n \in_R Z_q^*$ then computes

$$\begin{aligned} r'_n &= e(P,P)^{k_n} \\ r_n &= r_{n-1} r'_n \\ c_n &= h(m, r_n) \\ u_n &= u_{n-1} + c_n S_{ID_n} + k_n P \end{aligned}$$

Then he sends the final multisignature $\langle u_n, c_1, c_2, ..., c_n\rangle$ along with the message m to the verifier.

4.4 Multisignature Verification

On receiving a signature $\langle u_n, c_1, c_2, ..., c_n\rangle$ and message m, the receiver verifies the signature by computing

$$r_n = e(u_n, P)e(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub})$$

accepts the multisignature if and only if $c_n = h(m, r_n)$

5 Analysis

5.1 Correctness

The verification of the multisignature $\langle u_n, c_1, c_2, ..., c_n\rangle$ is justified by the following equations:

$$\begin{aligned} e(u_n, P)e(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}) &= e(\sum_{i=1}^{n}(c_i S_{ID_i} + k_i P), P)e(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}) \\ &= e(\sum_{i=1}^{n} c_i S_{ID_i}, P)e(\sum_{i=1}^{n} k_i P, P)e(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}) \\ &= e(\sum_{i=1}^{n} c_i Q_{ID_i}, P_{pub})e(\sum_{i=1}^{n} k_i P, P)e(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}) \\ &= e(\sum_{i=1}^{n} k_i P, P) \end{aligned}$$

$$= \prod_{i=1}^{n} e(P,P)^{k_i}$$
$$= \prod_{i=1}^{n} r'_i$$
$$= r_n$$

So, one can verify $c_n = h(m, r_n)$

5.2 Security

Here we first define the adversary for our scheme and then present the security analysis. The adversary for our serial multisignature scheme (SMS) is defined with following capabilities:
SMS-Adversary Given the system parameters `params` an adversary $\mathcal{A}_{\mathcal{SMS}}$, which can ask hash and signing queries, executes the following $\forall j \in [1, l]$ with given l.

- An SMS adversary $\mathcal{A}_{\mathcal{SMS}}$ selects a message m_j, the ith signer $\mathbf{S}_{i_j}$ and the signer's identity ID_{i_j}.
- Generates a valid partial multisignature $\langle u_{i_j-1}, c_{1_j}, c_{2_j}, ..., c_{i_j-1}\rangle$ by colluding with $\mathbf{S}\backslash\mathbf{S}_{i_j}$,
- Sends $\langle u_{i_j-1}, c_{1_j}, c_{2_j}, ..., c_{i_j-1}\rangle$ to $\mathbf{S}_{i_j}$
- Gets a valid partial multisignature $\langle u_{i_j}, c_{1_j}, c_{2_j}, ..., c_{i_j}\rangle$ from the $\mathbf{S}_{i_j}$.

We say that the SMS adversary $\mathcal{A}_{\mathcal{SMS}}$ is successful if after l iterations of these steps, it can compute a multisignature for a message m such that $m \neq m_j$ and at least one ID of the signers is not in ID_{i_j} $\forall j \in [1, l]$.

Theorem 1: The scheme proposed in [8] is secure against existential forgery under adaptive chosen message attack in the random oracle model.
Proof of the above theorem is also given in [8].
Theorem 2: The SMS is a secure serial multisignature scheme in the random oracle.
Proof: Let $\mathcal{A}_{\mathcal{SMS}}$ be a polynomial-time adversary for our SMS scheme and let $\mathcal{A}_{\mathcal{HS}}$ be a polynomial-time adversary for our base scheme [8]. Utilizing the result of Theorem 1, we prove that our SMS is secure.

The idea behind this proof is that if $\mathcal{A}_{\mathcal{SMS}}$ manages to frame an honest signer by constructing a valid multisignature on an arbitrary message without interacting with this honest signer, then $\mathcal{A}_{\mathcal{HS}}$ can forge a previously unsigned message. $\mathcal{A}_{\mathcal{HS}}$ can query the hash and signing oracles with an identity ID for any arbitrary message. Whenever $\mathcal{A}_{\mathcal{SMS}}$ wants to get a valid multisignature scheme by framing an honest signer, it sends the signing query to $\mathcal{A}_{\mathcal{HS}}$. Then, $\mathcal{A}_{\mathcal{HS}}$ forwards the signing query to its signing oracle using the identity and the partial multisignature given by $\mathcal{A}_{\mathcal{SMS}}$. $\mathcal{A}_{\mathcal{HS}}$ returns the reply back to $\mathcal{A}_{\mathcal{SMS}}$. It is easy to see that $\mathcal{A}_{\mathcal{SMS}}$ will be successful in its attempts if the reply from $\mathcal{A}_{\mathcal{HS}}$ is a valid signature. But, $\mathcal{A}_{\mathcal{HS}}$ generating a valid signature is a clear contradiction to the result of the Theorem 1. Hence the proof.

6 Conclusion

With Hess's ID-based signature scheme as the base, we have presented an ID-based serial multisignature scheme using bilinear pairings. Our scheme requires a forced verification at every level, which avoids the overlooking of the signatures of all the predecessors. Moreover, the verification cost does not increase exponentially like some of the existing multisignature schemes. To the best of our knowledge there is no existing secure serial ID-based multisignature scheme using pairings. We also proved that the scheme is secure against existential forgery under adaptive chosen message attack in the random oracle model.

References

1. Bellare, M., Rogaway, P.: Random Oracles are Practicle - a paradigm for designing efficient protocols. In: First ACM Conference and Communications Security, ACM, 1993 62-73
2. Boldyreva, A.: Threshold Signatures, Multi Signatures and Blind Signatures Based on the GDH group Signature Scheme. In: PKC 2003, Lecture Notes in Computer Science, Vol. 2567, Springer-Verlag, (2003) 31-46
3. Boneh, D., Franklin, M.: Identity Based Encryption from the Weil Pairing. SIAM Journal of Computing, , 32(3), (2003) 586-615. Extended abstract In: Proceedings of CRYPTO 2001, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, (2001) 213-229.
4. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Asiacrypt 2001, Lecture Notes in Computer Science, Vol. 2248, Springer-Verlag, (2002) 514-532.
5. Boyd, C.: Digital Multisignatures. Cryptography and Coding, Oxford University Press, (1989), 241-246.
6. Chen, X., Zhang, F., Kim, K.: ID-based Multi-Proxy Signature and Blind Multisignature from Bilinear Pairings. In: Proceedings of KIISC'2003, Korea (2003) 11-19.
7. Harn, L.: Group-oriented (t, n) threshold digital signature scheme and multisignature. In: IEE Proceedings - Computers and Digital Techniques, 141(5), (1994) 307-313.
8. Hess, F.: An Efficient Identity Based Signature Schemes Based on Pairings.In: SAC 2002, Lecture Notes in Computer Science, Vol. 2595, Springer-Verlag, (2002) 310-324.
9. Horster, P., Michels, M., Petersen, H.: Meta-Multisignatures Schemes Based on the Discrete Logarithm Problem. In: IFIP/Sec 1995, (1995) 128-142.
10. Itakura, K., Nakamura, K.: A Public Key Cryptosystem Suitable for Digital Multi Signatures. In: NEC Research and Development,(1983) 71:1-8.
11. Lin, C. Y., Wu, T. C., Hwang, J.: ID-based Structured Multi Signature Schemes. In: Advances in Network and Distributed Systems Security, Kluwer Academic publishers (IFIP Conference Proceedings 206), Boston (2001) 45-59.
12. Lin, C. Y., Wu, T. C., Zhang, F.: A Structured Multi Signature Scheme from the GDH group. In: Cryptology ePrint Archive, Report 2003/090, available at http://eprint.iacr.org/2003/090.
13. Micali, S., Ohta, K., Reyzin, L.: Accountable Subgroup Multi Signatures. In: ACM Conference on Computer and Communications Security, ACM,(2001) 245-254.
14. Mitchell, C. J.: An attack an ID-based nulti signature sheme. In: Royal Holloway, University of London, Mathematics Department Technical Report RHUL-MA-2001-9 (December 2001).
15. Ohata, T., Okamoto, T.: A digital multisignature scheme based on the FiatShamir scheme. In: Asiacrypt 91, Lecture Notes in Computer Science, Vol. 739, Springer-Verlag, (1991) 75-79.

16. Ohta, K., Okamoto, T.: Multi Signature Scheme Secure Against Active Insider Attacks. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, E82-A(1),(1999) 21-31.
17. Okamoto, T.: A Digital Multi Signature Schema Using Bijective Public Key Crypto Systems. ACM Transactions on computer systems, ACM, 6(4), (1988) 432-441.
18. Shamir, A.: ID-based Cryptosystems and Signature Schemes. In: Proceedings of Crypto 84, Lecture Notes in Computer Science, Vol. 196, Springer, (1985) 47-53.