Proceedings of International Conference
on Cryptology and Information Security

The Interactive Protocol for Digital Multisignature

Lein Harn
Department of Computer Networking
University of Missouri - Kansas City
Kansas City, MO 64110
TEL:(816)235-2367
FAX:(816)235-5159
Email:HARN@CSTP.UMKC.EDU

Abstract

A multisignature scheme has been proposed by Harn in 1994 which requires a broadcast channel in order to prevent signature forgery. This paper proposes an interactive protocol which allows all signers to reach a synchronous mode by generating commitments first and then each signer needs to verify the authenticity of all other signers' commitments before signing any individual signature. The protocol can be proceeded in any order. After all signers having signed the same message, a multisignature can be computed by the very last signer by combining all individual signatures. The length of multisignatures is always constant and does not depend on the number of signers involved. Anyone can verify this multisignature very efficiently by using the product of all signers' public keys.

1. Introduction

A digital signature is generated by an individual signer with the knowledge of a secret. Digital signature can be used to establish sender's authenticity and to resolve a dispute between the sender and the receiver. However, for some applications, there are multiple users involved in signing a message. This problem is the so-called 'multisignature' problem.Digital signature/multisignature is generated based on a one-way trap-door function. With the knowledge of a set of secrets, multiple signers work together to generate a one-way output as the digital multisignature.

The security of most digital multisignature schemes [1-4] which were proposed before 1994 is based on the factoring problem. According to [5], since each user selects an unique modulus n for its public key, there are two problems associated with this approach: (1) the signing order has certain restrictions and (2) the multisignature verification requires to process through all signers' moduli n. In 1994, L. Harn [5] has proposed an efficient multisignature scheme. The security of this proposed scheme is based on the

discrete logarithm problem. This scheme allows users to sign each individual signature separately. After all signers having signed the same message, a multisignature can be computed by a server. In addition, the length of this multisignature is always constant and does not depend on the number of signers involved. Anyone can verify this multisignature very efficiently by using the product of all signers' public keys. A number of group oriented systems based on Harn's approach have been proposed and the reader is referred to [6-10] for current development of multisignature schemes.

One major disadvantage of Harn's scheme is that a simultaneous broadcast channel is required in order to prevent signature forgery. There have some other threats to Harn's scheme which can be found in [11].

This paper proposes an interactive protocol for generating multisignatures. Each signer signs an individual signature separately. After all signers having signed the same message, a multisignature can be computed by the very last signer. There is no simultaneous broadcast channel needed and no special signing order is required. The multisignature signed by this protocol has the same property as that signed by the Harn scheme [5].

2. The Interactive Protocol for Digital Multisignature

There are two phases in the protocol to generate any multisignature. The first phase is the so-called 'commitment phase'. The objective of this phase is to enable all signers to reach a synchronous mode by generating commitments. The second phase is the so-called 'signing phase'. Each signer verifies the authenticity of all other signers' commitments and then signs an individual signature. With this additional commitment phase as compared with the Harn scheme [5], there is no need for a simultaneous broadcast channel in this protocol.

Throughout this paper, we assume that there are n signers with their identities ID_i, for $i=1, 2,...., n$, to sign the same message m, where m is formed by concatenating the original message with all signers' identities. Although the following protocol can be proceeded in any order among all signers and signing orders in two phases can be different, for the sake of convenience, we will assume that the proceeding order is $ID_1, ID_2, ..., ID_n$.

A large prime, p, a primitive element, α, of $GF(p)$, and a one-way function f need to be made public. Each signer randomly selects a secret integer z_i from $[1, p-1]$ and computes a corresponding public key as

$v_i = \alpha^{z_i} \mod p.$

$\{z_i, v_i\}$ are the secret and public keys for the commitment.

Similarly, each signer randomly selects a secret integer x_i from $[1, p-1]$ and computes a corresponding public key as

$y_i = \alpha^{x_i} \mod p.$

Then the public key y associated with multisignatures is determined as

$y = \prod_{i=1}^{n} y_i \mod p.$

We start with the multisignature-commitment phase.

Commitment Phase:

For $i=1$ to $n-1$ do

If the signer u_i agrees to sign the message m, the signer randomly selects a number k_i from $[1, p-1]$ to compute

$r_i = \alpha^{k_i} \mod p,$

and uses the secret key z_i to compute c_i to satisfy

$r_i z_i = k_i + c_i \mod \varnothing(p),$ (1)

$\{r_i, c_i\}$ is the individual commitment and $\{m, (r_j, c_j,$ for $j= 1, 2, ...,i)\}$ is sent to the next signer;

otherwise, the signer u_i stops the protocol and no multisignature can be generated;

End;

If the signer u_n agrees to sign the message m, the signer will generate $\{r_n, c_n\}$ in the same way and $\{m, (r_j c_j,$ for $j= 1, 2, ...,n)\}$ is sent to the signer u_1; otherwise the signer u_n stops the protocol.

Signing Phase:

For $i=1$ to $n-1$ do

Signer u_i uses all other signers' public keys v_j, for $j= 1, 2, ...,n, j \neq i$, to verify all other signers' commitments by computing whether

$v_j^{r_j} r_j \alpha^{c_j} \mod p$, for $j= 1, 2, ...,n, j \neq i.$ (2)

If commitments have been verified, the signer computes the value r as

$r = \prod_{j=1}^{n} r_j \mod p,$

signer u_i uses his/her secret keys, x_i and k_i, to compute

$s_i = x_i m' - k_i r \mod \varnothing(p),$ (3)

where $m' = f(m)$, $\{r_i, s_i\}$ is the individual signature of the message m, and $\{m, (r_j c_j,$ for $j= 1, 2, ...,n), (s_j,$ for $j=1, 2, ..., i)\}$ is sent to the next signer; otherwise, the signer stops the protocol and no multisignature can be generated;

End;

If the signer u_n verifies all other signers' commitments, the signer will generate s_n in the same way and compute

$s = s_1 + s_2 + ... + s_n \mod p-1,$

$(r, s\}$ is the multisignature for the message m;

otherwise, the signer u_n stops the protocol.

Multisignature Verification:

An outsider determines all signers' identities in m first and then needs to use all signers' public key, $y = \prod_{i=1}^{n} y_i \bmod p$, to verify the validity of the multisignature. The verification procedure is given as $y^{m'} = r^r \alpha^s \bmod p$, where $m' = f(m)$. According to [5], if the above equation holds true, the multisignature $\{r, s\}$ has been verified.

3. Security and Discussion:

Since the signature scheme in the signing phase is the same as that used in [5], we do not need to repeat the security analysis of the signature scheme. However, in this protocol, with the additional commitment phase, there is no need for a broadcast channel. We need to analyze the security of the commitment scheme.

(a) Since x_i, z_i and k_i are three secret numbers, it is infeasible to solve these secret numbers based on equations (1) and (3).

(b) According to equation (2), it is computational infeasible to forge a commitment $\{r_i, c_i\}$ without knowing the secret key z_i.

(c) In Harn's multisignature scheme [5], an attack is possible if it allows one designate server to compute the r for all signers. The server can, instead of sending a genuine r, send a computed r' to all signers to forge any signature. Thus, a broadcast channel is needed to allow each signer to compute this r. In fact, in this attack, the server will have no knowledge of the discrete logarithm of r'. Thus, in our proposed protocol, we include a commitment phase which allows each signer to authenticate that all other signers' commitments r_i are generated by legitimate signers with knowledge of discrete logarithms of r_i, for $i=1,2,....,n$.

(d) A valid multisignature requires all signers work incorporatively. In this protocol, if any signer fails to work incorporatively, a valid multisignature will not be generated and secret keys of signers are still kept secret.

(e) Generalized ElGamal-type signature schemes can be found in [12]. Since the commitment scheme and the signature scheme are all variations of the ElGamal signature scheme, we list all variations of commitment schemes in the following:

	Commitment Scheme	Commitment Verification
(1)	$r_i z_i = k_i + c_i \bmod \varnothing(p)$	$v_i^{r_i} = r_i \alpha^{c_i} \bmod p$
(2)	$c_i z_i = k_i + r_i \bmod \varnothing(p)$	$v_i^{c_i} = r_i \alpha^{r_i} \bmod p$
(3)	$z_i = r_i k_i + c_i \bmod \varnothing(p)$	$v_i = r_i^{r_i} \alpha^{c_i} \bmod p$
(4)	$z_i = c_i k_i + r_i \bmod \varnothing(p)$	$v_i = r_i^{c_i} \alpha^{r_i} \bmod p$

Similarly, we list all variations of signature schemes which can be used to

generate an efficient multisignature in the following:

Signature Scheme	*Signature Verification*
(1) $m'x_i=rk_i+s_i \bmod \varnothing(p)$	$y_i^{m'}=r_i^r \alpha^{s_i} \bmod p$
(2) $rx_i=m'k_i+s_i \bmod \varnothing(p)$	$y_i^r=r_i^{m'} \alpha^{s_i} \bmod p$
(3) $rm'x_i=k_i+s_i \bmod \varnothing(p)$	$y_i^{rm'}=r_i \alpha^{s_i} \bmod p$
(4) $x_i=m'rk_i+s_i \bmod \varnothing(p)$	$y_i=r_i^{m'r} \alpha^{s_i} \bmod p$
(5) $(r+m')x_i=k_i+s_i \bmod \varnothing(p)$	$y_i^{r+m'}=r_i \alpha^{s_i} \bmod p$
(6) $x_i=(m'+r)k_i+s_i \bmod \varnothing(p)$	$y_i=r_i^{m'+r} \alpha^{s_i} \bmod p$

(f) In above protocol, we assume that there are always n signers in generating multisignatures. In fact, the protocol works properly even when the number of signers is not a constant integer. Under this situation, in the commitment phase the initiator needs to prepare a draft of the message to be signed by a list of candidates. The protocol needs to be proceeded through all possible candidates. If any candidate agrees to become a signer, he/she needs to make a commitment as proposed in the protocol. Otherwise, he/she just forwards the message and the list to anyone in the list who has not responded to the protocol. In the signing phase, only committed signers need to generate individual signatures.

4. Conclusion

We have proposed an interactive protocol for generating multisignatures. There are two phases in this protocol. In the commitment phase, all signers synchronize themselves by generating commitments. In the signing phase, each singer verifies all commitments before signing any individual signature. This protocol can be proceeded in any order. The very last signer in this protocol can combine all individual signatures into a multisignature without any data expansion.

References

[1] C. Boyd. Digital multisignature. In *Conference on Coding and Cryptography,* Cirencester, 15-17 December 1986.

[2] T. Kiesler and L. Harn. RSA blocking and multisignature schemes with no bit expansion. In *Electronics Letters,* Vol. 26, No. 18, pp. 1490-1491, August, 1990.

[3] K. Ohta and T. Okamoto. A digital multisignature scheme based on the Fiat-Shamir scheme. In *Advances in Cryptology, Proc. of Asiacrypt '91,* Nov. 11-14, 1991.

[4] T. Okamoto. A digital multisignature scheme using bijective public-key cryptosystems. In *ACM Trans. on Comp. Systems,* Vol. 6, No. 8, pp. 432 -441, 1988.

[5] L. Harn. Group-oriented (t, n) threshold digital signature scheme

and digital multisignature. In *IEE Proc.-Comput. Digit Tech.*, Vol. 141, No. 5, pp. 307-313, Sep. 1994.

[6] C. Li, T. Hwang, and N. Lee. (t, n)-threshold signature scheme based on discrete logarithm. In *Pre-proc. of Eurocrypt '94*, pp. 191-200, May, 1994.

[7] S. K. Langford. Threshold DSS signatures without a trusted party. In *Proc. of Crypto '95*, pp. 397-409, Aug. 1995.

[8] J. Jiahui, and Z. Renjie. Digital Multisignature schemes based on the Schnorr scheme. In *Chinacrypt '96*, pp. 170-176, April 1996.

[9] L. Langru, Z. Renjie, and H. Lining. A (t, n) threshold group signature scheme. In *Chinacrypt '96*, pp. 177-184, April 1996.

[10] J. J.-R. Chen, and P.-T. Sun. A multisignature scheme based on discrete logarithm problem. In *Proc. of National Information Security Conference 1996, Republic of China*, pp. 61-69, May 1996.

[11] P. Horster, M. Michels, and H. Petersen. Meta-multisignature schemes based on the discrete logarithm problem. In *Proc. of IFIP/SEC '95*, Chapman & Hall, pp. 128-142, 1995.

[12] L. Harn, and Y. Xu. Design of generalized ElGamal type digital signature scheme based on discrete logarithm. In *Electronics Letters*, Vol. 30 No. 24, pp. 2025-2026, Nov. 1994.