Cover

An attack on a multisignature scheme

Zheng Dong, Kefei Chen

Department of Computer Science and Engineering

Shanghai Jiaotong University

Shanghai 200030, China

{zheng-dong,chen-kf}@cs.sjtu.edu.cn

In this letter, we show that structured ElGamal-type multisignature scheme due to Burmester *et al*. is not secure if the adversary attacks key generation.

Keywords: cryptanalysis, multisignature, authentication

An attack on a multisignature scheme

Zheng Dong, Kefei Chen

In this letter, we show that structured ElGamal-type multisignature scheme due to Burmester et al .is not secure if the adversary attacks key generation.

Introduction: Multisignature scheme realizes that plural users generate the signature on a message, and that the signature is verified. Recently, Burmester et al.[1] presented a structured ElGamal-type scheme (Burmester *et al.*'s scheme), which is based on discrete logarithm problem (DLP). This letter shows that the Burmester et al.'s scheme is not secure if the adversary attacks *key generation*. In the following, the brief review of Burmester et al.'s scheme is given, and then an attack is proposed.

Brief review of Burmester et al.'s scheme

We assume that n signers I_1 , I_2 ,…I_n generate a signature on a fixed message M according to order fixed beforehand.

Key generation: In their scheme, there are three public system parameters. The parameter p and q are two large prime numbers, $p > q$, the parameter $g \in Z_p^*$ is an element with order q. *h( )* is a public hash function. Each user selects his private key $a_i \in Z_q^*$, then computes his public key sequentially as follows: $y_1 = g^{a_1} (\bmod p)$, $y_i = (y_{i-1} \cdot g)^{a_i} (\bmod p)$, then a public key of ordered group (I_1 , I_2 ,…I_n) is set to $y = y_n$.

Signature generation:

(1) Generation of *r*: signer I_1 , I_2 ,…I_n generate *r* together as follows:

1. Player I_1 selects $k_1 \in Z_q^*$ randomly and computes $r_1 = g^{k_1} \bmod p$. If $\gcd(r_1, q) \neq 1$, then select new k_1 again.

2. For $i \in \{2,...,n\}$, a signer I_{i-1} sends r_{i-1} to I_i . And I_i selects $k_i \in Z_q^*$ randomly and computes $r_i = r_{i-1}^{a_i} \cdot g^{k_i} (\bmod p)$. If $\gcd(r_i, q) \neq 1$, then select new k_i again.

3. $r = r_n$

(2) Generation of *s*: Signer I_1, I_2 ,…I_n generate *s* together as follows:

1. I_1 computes $s_1 = a_1 + k_1 \cdot r \cdot h(r, M) \bmod q$

2. For $i \in \{2,...,n\}$; I_{i-1} sends s_{i-1} to I_i. I_i verifies that $g^{s_{i-1}} \stackrel{?}{=} y_{i-1} r_{i-1}^{r \cdot h(r,M)} \bmod p$, then computes $s_i = (s_{i-1} + 1) a_i + k_i \cdot r \cdot h(r, M) \bmod q$

3. $s = s_n$

(3) The multisignature on *M* by order (I_1, I_2,…I_n) is given by (*r*, *s*).

Signature verification:

A multisignature (*r*, *s*) on *M* is verified by $g^s \stackrel{?}{=} y_{i-1} r^{r \cdot h(r,M)} \bmod p$.

If the adversary attacks key generation, the above scheme is not secure at all.

Our attack

Key generation is the same as Burmester *et al.*'s scheme but that player I_j is bad and generates his public key by choosing a secret key $a_j \in Z_q^*$ and setting $y_j = g^{a_j} (\bmod p)$. The key of ordered group ($I_1, I_2, \ldots, I_n$) is set to $y = y_n$

In this case, The multisignature (*r, s*) on *M* can be generate without $I_1, \ldots, I_{j-1}$ signing it:

(1) Generation of *r*:

1. Player I_j selects $k_j \in Z_q^*$ randomly and computes $r_j = g^{k_j} \bmod p$. If $\gcd(r, q) \neq 1$, then select new k_j again.

2. for $i \in \{j+1, \ldots, n\}$, a signer I_{i-1} sends r_{i-1} to I_i. And I_i selects $k_i \in Z_q^*$ randomly and computes $r_i = r_{i-1}^{a_i} \cdot g^{k_i} (\bmod p)$. If $\gcd(r_i, q) \neq 1$, then select new k_i again.

3. $r = r_n$

(2) *Generation of s:* signer $I_1, I_2, \ldots I_n$ generate *s* as follows:

1. I_j computes $s_j = a_j + k_j \cdot r \cdot h(r, M) \bmod q$

2. for $i \in \{j+1, \ldots, n\}$, I_{i-1} sends s_{i-1} to I_i. I_i verifies that $g^{s_{i-1}} \stackrel{?}{=} y_{i-1} r_{i-1}^{r \cdot h(r,M)} \bmod p$, then computes $s_i = (s_{i-1} + 1) a_i + k_i \cdot r \cdot h(r, M) \bmod q$

3. $s = s_n$

The bad multisignature on *M* is (*r, s*)

Verification:

The following equation is still hold

$$g^s = y \cdot r^{r \cdot h(r,M)} \bmod p$$

The above attack shows that I_j can cheat I_{j+1}, …, I_n to sign any message M without knowing I_1,…, I_{j-1} not signing it. Especially, when $j = n$, player I_j can sign any message M it wants on behalf of the entire group $\{I_1, I_2, \ldots, I_n\}$.

Conclusion: we have presented an attack on Burmester et al.'s scheme, the attack shows that Burmester *et al.*' scheme is insecure against attacks on key generation. It is possible to modify the Burmester *et al.*'s scheme by requiring that each player I_i to provide a zero-knowledge proof of knowledge (ZKPoK) of the discrete log of y_i / y_{i-1} in base g.

Acknowledgment: This work was partially supported by NSFC under grants 90104005, 60173032 and 60273049.

References

[1] M. Burmester, Yvo Desmedt, Hiroshi Doi, Masahiro Mambo, Eiji Okamoto, Mitsure Tada, and Y. Yoshifuji, "a structured ElGamal-Type multisignature scheme", Advances in Cryptology-Proceedings of PKC'2000, Lecture notes in computer science,(2000), Spfinger-Verlag, 466-482.

Authors' affiliations:

Zheng Dong, Kefei Chen (Department of Computer Science and Engineering, Shanghai Jiaotong University, 1954 Hua Shan Road, Shanghai 200030, People's Republic of China)

Email: {zheng-dong, chen-kf }@cs.sjtu.edu.cn