A Digital Multisignature Scheme with Distinguished Signing Responsibilities

Constantin Popescu

University of Oradea

Department of Mathematics

Str. Armatei Romane 5, Oradea

ROMANIA

E-mail: cpopescu@uoradea.ro

Abstract: Digital multisignature is signed by multiple signers with the knowledge of multiple secret keys and can be verified based on all signers' public keys. An efficient digital multisignature scheme can combine multiple individual signatures of the same message into a single multisignature and this multisignature can be verified efficiently. In this paper we describe a multisignature scheme with distinguished signing responsibilities. In this scheme each group member has distinguished signing responsibility and partial contents of the message can be verified without revealing the whole message.

Keywords: multisignature scheme, distinguished signing responsibilities, elliptic curve.

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. In 2001 he received the Ph. D degree in informatics (cryptography) at the Babes-Bolyai University, Cluj Napoca. Since 2003 he has been Associate Professor at the Department of Mathematics, University of Oradea. His current research interests include cryptography, network security, group signatures, identification schemes.

1. Introduction

In [1] Harn has proposed two ElGamal type variants that can combine all individual signatures into a multisignature without any data expansion. In other words, the length of the multisignature is equivalent to the length of each individual signature. This result is reasonable since the length of the signature or multisignature depends only on the security parameters of signature schemes and not on the number of signers involved. Multiple signers with knowledge of multiple secret keys can produce a fixed length of digital signature. Let us summarize properties associated with these multisignature schemes:

1. The length of multisignature is fixed. This property minimizes the communication and memory costs of multisignatures.

2. The multisignature can be verified at once instead of verifying each one individually. This property speeds up the verification process significantly.

3. The public key associated with this multisignature is just the product of all individual public keys. This property reduces the size of public key directory since it needs only to store each signer's public key.

The application of digital multisignature can be found in some secret sharing applications. For example, a company's policy may require multiple managers to sign any business contract. Digital multisignature scheme enables this internal policy effectively. Each manager has to use his individual secret key to sign the same document and all individual signatures can be combined into a single multisignature. However, to any external verifier, this multisignature is just a normal signature that can be verified by using the company's public key, which is a product of all public keys of the signers. In the multisignature schemes proposed in [1], all group members hold the same responsibility of signing the document. In fact, there are some applications that need to use multisignatures with distinguished signing responsibilities. For example, a company releases a document that may involve financial department, engineering department and program office. Each entity is responsible of preparing and signing a particular section of the document. The signing responsibility of engineering department may have no interest to read the content prepared by the financial department. However, the combination of all sections represents the company's document. The company's document should be easily verified by any outsider using company's public key. For the sake of confidentiality, same verifier may be restricted to access and verify only some sections of the document. In this paper, we modify the schemes from

[1] to convert them to be a multisignature scheme with distinguished signing responsibilities. A multisignature scheme with distinguished signing responsibilities can be found in many cryptographic applications. For example, credit card, medical insurance and telephone companies can establish a joint venture to issue smart cards to customers. By using the multisignature proposed in this paper can:

1. Allow each company to register and sign its own customers.
2. Reduce memory storage of each card since only a multisignature is needed on each card.
3. Reduce memory storage of each verifier since only the group public key is needed.
4. Enhance the card security since multiple private keys needed to forge a multisignature.
5. Speed up signature verification.

The multisignature scheme described in this paper has been described in the setting of the group of points on an elliptic curve defined over a finite field. Suitable choices include the multiplicative group of a finite field, subgroups Z_n^*, where n is a composite integer, and subgroups of Z_q^* of prime order q. Elliptic curve groups are advantageous because they offer equivalent security as the other groups but with smaller key sizes and faster computation times.

2. Digital Multisignature Schemes Using Elliptic Curves

In this section we describe the elliptic curve version of the multisignature scheme proposed in [1]. We assume that there are t signers U_i, $1 \le i \le t$ to sign the same message $m \in \{0,1\}^*$.

2.1 Key Generation

Many researchers have examined elliptic curve cryptosystems, which were firstly proposed by Miller [5] and Koblitz [2]. The elliptic curve cryptosystems which are based on the elliptic curve logarithm over a finite field have some advantages than other systems: the key size can be much smaller than the other schemes since only exponential-time attacks have been known so far if the curve is carefully chosen [3], and the elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm are broken.

Firstly, we choose elliptic curve domain parameters (see [8]):

1. Choose p a prime and n an integer. Let $f(x)$ be an irreducible polynomial over $GF(p)$ of degree n, generating finite field $GF(p^n)$ and assume that α is a root of $f(x)$ in $GF(p^n)$.
2. Two field elements $a,b \in GF(p^n)$, which define the equation of the elliptic curve E over $GF(p^n)$ (i.e., $y^2=x^3+ax+b$ in the case $p>3$, where $4a^3+27b^2 \neq 0$).
3. Two field elements x_p and y_p in $GF(p^n)$, which define a finite point $P=(x_p,y_p)$ of prime order q in $E(GF(p^n))$ ($P \neq O$, where O denotes the point at infinity).
4. The converting function $c(x): GF(p^n) \rightarrow Z_{p^n}$ which is given by

$$c(x)=\sum_{i=0}^{n-1} c_k p^i \in Z_{p^n}, \qquad x=\sum_{i=0}^{n-1} c_k \alpha^i \in GF(p^n),\ 0 \le c_i < p.$$

For the elliptic curve over finite field see more details in [4], [7].

The operation of the key generation is as follows:

1. Each signer randomly selects an integer d_i from the interval $[1,q-1]$ and computes a corresponding public key as the point:
$Q_i=d_iP$.
2. Compute the public key Q for all signers, which is equal to the sum of all individual public keys
$Q=Q_1+Q_2+\ldots+Q_t=dP=(x_Q,y_Q)$,
where
$d=d_1+d_2+\ldots+d_t \pmod q$.
3. Let H be a one-way hash function such as SHA-1 [6].

2.2 Generating the Multisignature

Each signer U_i, $1 \le i \le t$ executes next steps:

1. Randomly selects a number $k_i \in [1, q-1]$ and computes $R_i = k_i P = (x_{R_i}, y_{R_i})$, $1 \le i \le t$.
2. Converting the x-coordinate of point R_i into the integer $r_i = c(x_{R_i})$, where $c(x)$ is the converting function. The values r_i is broadcast to the other signer.
3. Once r_i, $1 \le i \le t$, are available through the broadcast channel, each signer computes the commitment r as

$r = r_1 + r_2 + \ldots + r_t \pmod q$.

4. Uses his secret keys, d_i and k_i, to sign the message m. The signer U_i computes

$s_i = d_i H(m) - k_i r \pmod q$.

5. Transmits the pair (m, s_i) to the clerk.

Once the clerk receives the individual signature (r_i, s_i) from U_i, he needs to verify the validity of this individual signature. The verification procedure is to compute the point

$(r^{-1} H(m) \bmod q) Q_i - (r^{-1} s_i \bmod q) P = (x_{e_i}, y_{e_i})$, $1 \le i \le t$

and check

$r_i = c(x_{e_i}) \pmod q$, $1 \le i \le t$.

Once all individual signatures are received and verified by the clerk, the multisignature of the message m can be generated as (r, s), where

$s = s_1 + s_2 + \ldots + s_t \pmod q$.

2.3 Verifying the multisignature

Since individual signatures (r_i, s_i), $1 \le i \le t$, satisfy

$(r^{-1} H(m) \bmod q) Q_i - (r^{-1} s_i \bmod q) P = (x_{e_i}, y_{e_i})$, $1 \le i \le t$.

Adding the above equations from 1 through t, we obtain

$(r^{-1} H(m) \bmod q) Q - (r^{-1} s \bmod q) P = (x_e, y_e)$.

where $s = s_1 + s_2 + \ldots + s_t \pmod q$, $Q = Q_1 + Q_2 + \ldots + Q_t = dP = (x_Q, y_Q)$ and $r = c(x_e) \pmod q$. In other words, the verifier computes the point (x_e, y_e) and check if $r = c(x_e) \pmod q$. If this is true, then (r, s) is accepted as the valid multisignature of the message m signed by the users U_i, $1 \le i \le t$.

3. Multisignature Scheme with Distinguished Signing Responsibilities

In this section we describe the proposed multisignature scheme with distinguished signing responsibilities. The elliptic curve domain and the key generation are the same as in Section 2.

3.1 Generating the multisignature

We assume that there are t signers U_i, $1 \le i \le t$. Instead of signing the same message $m \in \{0,1\}^*$ directly, each signer should prepare a section of message m_i that he is responsible of and broadcast $H(m_i)$ to all other signers, where H is the one way hash function.

The operation of generating the multisignature with distinguished signing responsibilities is as follow:

1. The signer U_i, $1 \le i \le t$, randomly selects a number $k_i \in [1, q-1]$ and computes
$R_i = k_i P = (x_{R_i}, y_{R_i})$, $1 \le i \le t$.

2. The signer U_i, $1 \le i \le t$, converting the x-coordinate of the point R_i into the integer $r_i = c(x_{R_i})$, where $c(x)$ is the converting function. The values r_i is broadcast to the other signer.

3. Once r_i, $1 \le i \le t$ are available through the broadcast channel, each signer U_i computes the value r as

$r = r_1 + r_2 + \ldots + r_t \pmod q$.

4. Each signer U_i, $1 \le i \le t$, uses his secret keys, d_i and k_i, to sign the message $M = H(H(m_1), H(m_2), \ldots, H(m_t))$, where $H(H(m_1), H(m_2), \ldots, H(m_t))$ means the hash value of the concatenation $of\ H(m_1), H(m_2), \ldots, H(m_t)$. The signer U_i computes

$s_i = d_i M - k_i r \pmod q$

and transmits the pair (M, s_i) to the clerk.

The clerk needs to verify the validity of the individual signature (r_i, s_i) from U_i. The verification procedure is to compute

$(r^{-1} M \bmod q) Q_i - (r^{-1} s_i \bmod q) P = (x_{e_i}, y_{e_i})$, $1 \le i \le t$.

and check

$r_i = c(x_{e_i}) \pmod q$, $1 \le i \le t$.

The multisignature of the message $m = (m_1, m_2, \ldots, m_t)$ is the pair (r,s), where $s = s_1 + s_2 + \ldots + s_t \pmod q$. Since each signer is responsible of preparing a section of message m, the pair (r,s) is a digital multisignature with distinguished signing responsibilities.

3.2 Verifying the multisignature

The verifier computes the point

$(r^{-1} M \bmod q) Q - (r^{-1} s \bmod q) P = (x_e, y_e)$, $1 \le i \le t$.

where $s = s_1 + s_2 + \ldots + s_t \pmod q$, $Q = Q_1 + Q_2 + \ldots + Q_t = dP$. Also, the verifier check if $r = c(x_e) \pmod q$. If this equality holds, the pair (r,s) is a digital multisignature with distinguished signing responsibilities of the message m.

4. Conclusions

In this paper we proposed a digital multisignature with distinguished signing responsibilities. Instead of signing the message $H(m_1, m_2, \ldots, m_t)$, each signer needs to sign the message $M = H(H(m_1), H(m_2), \ldots, H(m_t))$. The computation of $H(H(m_1), H(m_2), \ldots, H(m_t))$ is faster than that of $H(m_1, m_2, \ldots, m_t)$ because each signer needs only to compute his own $H(m_i)$ and the other $H(m_j)$, $j \ne i$, $1 \le i, j \le t$, has been computed by the other signer.

In the case some verifies only allowed to access partial contents of the message, the partial contents can still be verified using the group public key without revealing whole message. This feature can be achieved by just providing the one way hash values of the inaccessible contents to the verifier.

REFERENCES

1. ***National Institute of Standards and Technology, **Secure Hash Standard (SHS)**, FIPS Publication 180-1, April 1995.
2. L. HARN, **Group-oriented *(t,n)* Threshold Signature and Multisignature**, IEE Proceedings Computers and Digital Techniques, Vol. 141, No.5, pp. 307-313, 1994.
3. N. KOBLITZ, **Elliptic curve cryptosystems**, Mathematics of Computation, Vol. 48, pp. 203-209, 1987.
4. N. KOBLITZ, **CM-Curves with Good Cryptographic Properties**, Proceedings of Crypto'91, Lecture Notes in Computer Science, Springer-Verlag, pp. 279-287, 1992.
5. A. MENEZES, **Elliptic Curve Public Key Cryptosystems**, Kluwer Academic Publishers, 1993.
6. V. MILLER, **Uses of elliptic curves in cryptography**, In Advances in Cryptology, Proceedings of Crypto'85, Lecture Notes in Computer Sciences, Springer-Verlag, pp. 417-426, 1986.
7. V.V. PATRICIU, M. PIETROŞANU, I. BICA, C. CRISTEA, **Securitatea informatică în Unix şi Internet**, Editura Tehnică, 1998.
8. C. POPESCU, **Blind Signature and Blind Multisignature Schemes Using Elliptic Curves**, Studia Univ."Babes-Bolyai" Informatica, vol.XLIII, no. 2, pp. 43-49, 1999.