A Novel Threshold Scheme based on Elliptic Curve Cryptosystem and Grey System Theory

Tzer-Shyong Chen and Chieh-Yen Hsu
Department of Information Management, Tunghai University
Department of Communication Engineering, Da-Yeh University
E_Mail: arden@thu.edu.tw

Abstract

Based on elliptic curve cryptosystem, this paper proposes a scheme for communication between two groups. The scheme uses the \((t, n)\) threshold scheme for identity verification between groups, and the \((r, s)\) digital signature scheme for verification between groups and groups. The generation of group secret key is based on data generation. The variety of key generation makes the attack on keys difficult. The system mainly hopes to deal with on-line conference security among many people. It can be applied to identity verification for international trade between corporations or conferences between nations or political parties.

Keywords: Elliptic curve cryptosystem, group communication, \((t, n)\) threshold, group signature, group verification

1. Introduction

Since the introduction of elliptic curve into cryptography in 1987, many elliptic curve cryptosystems have been proposed. One of them is the famous ECDSS, which is the elliptic curve version of DSA. Another DSA-like system is proposed by Poupard and Stern in
In this paper, we propose a mechanism that combines ECDSS with the secret-sharing $(t, n)$ threshold scheme proposed by Shamir and Blakley in 1979. A $(t, n)$ threshold scheme is determined by two parameters: the threshold value $t$ and the number of member $n$.

The group communication of elliptic curve cryptosystem described in this paper satisfies the following characteristics:

Suppose $A$ and $B$ are two groups. $A = \{A_1, A_2, \ldots, A_p\}$ and $B = \{B_1, B_2, \ldots, B_q\}$, where $A_1, A_2, \ldots, A_p$ are members of $A$, and $B_1, B_2, \ldots, B_q$ are members of $B$. Suppose group $A$ enciphers plaintext $P_m$ to get ciphertext $C_m$ and sends $C_m$ to group $B$. For $A$ to encipher $P_m$, at least $t$ members of group $A$ must participate in generating $A$’s secret key for the encipherment. With fewer than $t$ members from group $A$, the message cannot be enciphered. Similarly, when group $B$ receives ciphertext $C_m$, at least $t$ members of group $B$ must get together to generate $B$’s secret key. With fewer than $t$ members, the message cannot be reconstructed.

The paper will describe methods for verification of members within the same group, methods for generating group secret key and group public key, and methods for generating group signature.

The paper is organized as follows: Section 2 explains the concepts of data generation and model generation. Section 3 describes how those concepts are applied in group cryptosystems. Section 4 is the conclusion of the paper.

### 2. Data Generation and Model Generation

#### 2.1 Data Generation

Let $X$ be a sequence of data constituted by positive integers:

$$ X = \{x_1, x_2, \ldots, x_n\} $$  \hspace{1cm} (2.1)

and $X^{(i)}$ be the 1-AGO (D.J.Long, 1989) generation of $X$:

$$ X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\} $$  \hspace{1cm} (2.2)

If $\forall x_i^{(i)} \in X^{(i)}$,

$$ x_i^{(i)} = \sum_{j=1, j \neq i}^{n} x_j $$  \hspace{1cm} (2.3)
A novel threshold scheme

then $X^{(i)}$ satisfies the differential equation

$$\frac{dX^{(i)}}{dt} + aX^{(i)} = u.$$  (2.4)

The solution to equation (2.4) is

$$x^{(i)}_{k+1} = (x^{(i)}_1 - \frac{u}{a})e^{-ak} + \frac{u}{a}$$  (2.5)

where $a$ and $u$ satisfy the following equations

$$\begin{pmatrix} a \\ u \end{pmatrix} = (B^TB)^{-1} B^T \hat{Y}_N$$  (2.6)

$$B = \begin{bmatrix} -\frac{1}{2}(x^{(i)}_1 + x^{(i)}_2) & 1 \\ -\frac{1}{2}(x^{(i)}_2 + x^{(i)}_3) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}(x^{(i)}_{n-1} + x^{(i)}_n) & 1 \end{bmatrix}$$  (2.7)

$$\hat{Y}_N = (x_2, x_3, \ldots, x_n)^T$$  (2.8)

Equations (2.1) to (2.8) with explanations can be found in (D.J.Long, 1989). If we only take the first term of (2.5), the equation is

$$x^{(i)}_{k+1} = (x^{(i)}_1 - \frac{u}{a})e^{-ak}$$  (2.9)

Equation (2.9) is called the half-solution of (2.4). For $k = 1, 2, \ldots, m$, we get the following sequence from (2.9):

$$\bar{X} = (ae^{-a}, ae^{-2a}, \ldots, ae^{-ma})$$  (2.10)

where $ae^{-ja} = INT(ae^{-ja})$, $ae^{-ja} \in \bar{X}$. Equation (2.10) is called the sequence of data generation of (2.1).

### 2.2 Model Generation

Suppose that the sequence of data generation of (2.1) is as follows:

$$\left( ae^{-a}, ae^{-2a}, \ldots, ae^{-ma} \right)$$  (2.11)

and $m$ is big enough. Let $p$ be a prime number. If we take modulo $p$ for each term $ae^{-ja}$ in (2.11) as follows
\[ \beta_j = \alpha e^{-j\alpha} \mod p \]  

we obtain the following set:

\[ \beta = \{\beta_1, \beta_2, \ldots, \beta_m\} . \]  

3. Group Cryptosystem

Groups A and B are the two sides of secret communication. \( P_m \) is the plaintext. \( C_m \) is the ciphertext. Group A enciphers \( P_m \) to get \( C_m \) and sends it to group B. Group B receives \( C_m \) and deciphers it to get \( P_m \). Both group A and group B use the elliptic curve cryptosystem

\[ E_p(a,b) : y^2 = x^3 + ax + b \mod p \]  

where \( a, b \in \mathbb{N}^+ \), \( D = 4a^3 + 27b^2 \mod p \neq 0 \). \( p \) is a large prime. \( G \) is the basic point over elliptic curve. \( G \in E_p(a,b) \).

3.1 Group A's Preparation

3.1.1 Certification Authority (CA) generates a secret key \( a_0 \) for group A

CA chooses \( X^A = \{x_1^A, x_2^A, \ldots, x_n^A\} \) and uses data generation to obtain

\[ (\alpha e^{-a}, \alpha e^{-2a}, \ldots, \alpha e^{-m-1}a, \alpha e^{-ma}) \]  

where \( \alpha e^{-ja} = INT(\alpha e^{-ja}) \).

By taking modulo \( p \) for each term in (3.2) and letting

\[ \beta_j = \alpha e^{-ja} \mod p \]  

we get the following set:

\[ \beta^A = \{\beta_1^A, \beta_2^A, \ldots, \beta_n^A\} \]  

CA then randomly chooses \( a_0 = \beta_j^A \in \beta^A \) as A's secret key.
3.1.2 \( CA \) chooses a secret interpolation polynomial for group \( A \) \( (a_{i-1}, \ldots, a_0 \in N) \):
\[
P_A(x) = a_{i-1}x^{i-1} + \cdots + a_1x + a_0
\]  
(3.5)

3.1.3 \( CA \) generates a secret key for each member of group \( A \):
\[
P_A(ID_i) = n_A,
\]
\[
P_A(ID_2) = n_A,
\]
\[
P_A(ID_3) = n_A,
\]  
(3.6)
\[
P_A(ID_4) = n_A,
\]
\[
P_A(ID_5) = n_A
\]

3.1.4 \( CA \) verifies the membership of a member in group \( A \):
Step1: \( CA \) chooses a random integer \( X \), computes
\[
Y = XG
\]  
(3.7)
and sends \( Y \) to member \( A_i \) of group \( A \).
Step2: \( A_i \) receives \( Y \), uses his secret key to compute
\[
C' = n_A(XG)
\]  
(3.8)
and sends \( C' \) back to \( CA \).
Step3: \( CA \) checks to see if \( C = nA_i(XG) = C' \). If it is true, the membership verification is successful.

**Fig 1.** Member Verification
(within the same group)
3.1.5 Generation of group $A$'s secret key and public key:

Suppose $CA$ receives group member $A_i$'s request. If the membership verification is successful, $CA$ generates order pairs $(x_i, y_i) = (ID_i, n_A)$:

$$(1, n_A), (2, n_A), \ldots, (t, n_A), \ldots$$

The generation of group secret key needs any $t$ ordered pairs. With fewer than $t$ pairs, the secret key center cannot generate the group secret key. With at least $t$ pairs from (3.9), the center can generate the secret key and public key of group $A$ using Lagrange interpolation polynomial $P_a(x)$:

$$P_a(x) = \sum_{j=1}^{t} y_j \prod_{i,j=1 \atop i \neq j}^{t} \frac{x - x_i}{x_j - x_i} \mod p$$

$$= a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1x + a_0 \mod p$$

Evaluate the polynomial at $x = 0$. The group secret key is $K_A = P_A(0) = n_A = a_0$ and the group public key is $P_A = K_AG = n_AG$.

3.2 Group $B$'s preparation

3.2.1 $CA$ generates a secret key $b_0$ for group $B$

$CA$ chooses $x^B = \{x_1^B, x_2^B, \ldots, x_m^B\}$ and uses methods similar to $A$'s to obtain the set

$$\beta^B = \{\beta_1^B, \beta_2^B, \ldots, \beta_m^B\}$$

$CA$ then randomly chooses $b_0 = \beta_1^B \in \beta^B$ as $B$'s secret key.

3.2.2 $CA$ chooses a secret interpolation polynomial for group $B$ ($b_i, i = 1, \ldots, b_0 \in \mathbb{N}$):

$$P_B(x) = b_{t-1}x^{t-1} + \cdots + b_1x + b_0$$

3.2.3 $CA$ generates a secret key for each member of group $B$:

$$P_B(ID_1) = n_{B_1},$$

$$P_B(ID_2) = n_{B_2},$$

$$P_B(ID_3) = n_{B_3},$$

$$(3.13)$$
A novel threshold scheme

\[ P_B(ID_A) = n_B, \]
\[ P_B(ID_A) = n_B. \]

3.2.4 CA verifies the membership of a member in group B:

Use method similar to what was described in Section 3.1.4.

3.2.5 Generation of group B’s secret key and public key.

Using methods similar to group A’s, the secret key and public key of group B are:

\[ K_B = P_B(0) = n_B \]
\[ P_B = n_BG, \]

respectively.

3.3 Group Encipherment

3.3.1 Group A enciphers \( P_m \) to get \( C_m \) and sends \( C_m \) to group B

Step1: Group A chooses a random integer \( k \). Using plaintext \( P_m \) and group B’s group public key \( P_B \), A computes

\[ C_m = \{kG, P_m + kP_B\} \]
\[ = \{kG, P_m + k(P_B(0)G)\} \]  \hspace{1cm} (3.14)

Step2: Group A sends \( C_m \) to group B.

3.3.2 Group B receives \( C_m \) and deciphers \( C_m \) to get \( P_m \).

B deciphers \( C_m \) by computing the difference between \( C_m \)'s second term \( C_{m2} = P_m + kP_B \)
and the product of \( C_m \)'s first term \( C_{m1} = kG \) and B's group secret key \( n_B \):

\[ C_{m2} - n_BC_{m1} = P_m + kP_B - n_B(kG) \]
\[ = P_m + kP_B - k(n_BG) \]
\[ = P_m \]  \hspace{1cm} (3.15)

3.4 Group Signature Verification

3.4.1 System parameters

1. \( E_p(a,b) : y^2 = x^3 + ax + b \mod p, \)
\[ D = 4a^3 + 27b^2 \mod p \neq 0 \]

2. Hash function \( h(x) \), signature plaintext \( m \)
3.4.2 Group $A$ computes $(r, s)$ and sends $(r, s)$ to $B$.

Step 1: Group $A$ chooses a random integer $t$ as the session key, where $t \in [1, n-1]$, $t \neq n_A$, and $t \neq n_A$. Then compute

$$R = (x_R, y_R) = tP$$

$$r = x_R \mod n$$

$$s = t + h(m) + dr \mod n$$

Here, $h$ is a hash function, $r, s \neq 0$, $d = k_A = h_A(0)$, $h(0) \in [1, n-1]$.

Step 2: Group $A$ sends $(r, s)$ to $B$.

3.4.3 Group $B$ verifies $(r, s)$

Step 1:

$$u_1 = s - h(m) \mod n$$

$$u_2 = r \mod n$$

$$R' = (x'_R, y'_R) = u_1P - u_2Q$$

$$r' = x'_R \mod n$$

Step 2: $B$ checks to see if

$$r = x_R \mod n = x'_R \mod n = r'$$

If it is true, the verification is successful.

4. Conclusion

Elliptic curves have played an increasingly important role in many cryptographic situations. One of their advantages is that they offer a level of security comparable to classical cryptosystems (such as RSA) that use much larger key sizes. The security of elliptic curve cryptosystems is based on the elliptic curve discrete logarithm problem. There is no good general attack on this problem.

Through membership verification of members among the same group, the problem concerning counterfeit from the outsiders or members in the same group can be solved. On the other hand, group signature and group verification solves the dispute that arises between
A novel threshold scheme

two groups on the authenticity of a group's signature. For group secret key generation, the variety of \( n_a \) and \( n_b \) makes the attack on secret key difficult. If the secret key is lost, with \( X^A \) and \( X^B \), the original key can be regenerated. In so doing, we get a more reliable system that may work properly even if the secret key is lost.

References


Received: October 13, 2005