A Directed Threshold Signature Scheme with Threshold Verification

Sunder Lal * and Manoj Kumar **

* Dept of Mathematics,
IBS Khandari, Dr. B. R. A. University Agra.
Sunder_lal2@rediffmail.com.in.

**Department of Mathematics,
R.K. (P.G.) College Shamli,
C. C. S. University Meerut,
Utter Pradesh- India- 247776.
E-mail- yamu_balyan@yahoo.co.in

ABSTRACT

This paper proposes a signature scheme where the signatures are generated by the cooperation of a number of people from a given group of senders and the signatures are verified by a certain number of people from the group of recipients. Shamir’s threshold scheme and Schnorr’s signature scheme are used to realize the proposed scheme.

KEYWORDS

Threshold signature scheme, Lagrange interpolation, ElGamal Public key Cryptosystem and Threshold verification.

1. INTRODUCTION

Physical signature is a natural tool to authenticate the communication, but it is useless in electronic messages; one has to rely on other methods like digital signature. Digital signature is a cryptographic tool to solve this problem of authenticity in electronic communications. Basically digital signature has a property that anyone having a copy of the signature can check its validity by using some public information, but no one else can forge the signature on another document. This property of digital signature is called self-authentication. In most situations, the signer and the verifier is generally a single person. However when the message is sent by one organization to another organization, a valid message may require the approval or consent of several people. In this case, the signature generation and verification is done by more than one consenting rather than by a single person. A common example of this policy is a large bank transaction, which requires the signature of more than one person. Such a policy could be implemented by having a separate digital signature for every required signer, but this solution increases the effort to verify the message linearly with the number of signer. Threshold signature is an answer to this problem.

The (t, n) threshold signature schemes [1,2,3,5,6,8] are used to solve these problems. Threshold signatures are closely related to the concept of threshold cryptography, first introduced by Desmedt [1,2]. In 1991 Desmedt and Frankel [1] proposed the first (t, n) threshold digital signature scheme based on the RSA assumption. This paper proposes a Threshold signature scheme with threshold verification based on Shamir’s threshold signature scheme [10] and Schnorr’s signature scheme [9]. These basic tools are briefly described in the next section. The section-2 presents some basic tools. In Section-3 we present a Threshold Signature Scheme with threshold verification. Section-4 discusses the security of the Scheme. An illustration to the scheme is discussed in section-5. Remarks are in section-6.

2. PRELIMINARIES

Throughout this paper, we will use the following system setting.

• A prime modulous p, where $2^{511} < p < 2^{512}$;
• A prime modulous q, where $2^{159} < q < 2^{160}$ and q is a divisor of $p - 1$;
• A number g, where $g \equiv k (p - 1)/q \mod p$, k is random integer with $1 \leq k \leq p - 1$ such that g > 1; (g is a generator of order q in $\mathbb{Z}_p^*$).
• A collision free one-way hash function $h$ [25];

The parameters p, q, g and h are publicly known. Every user has two keys one private and one public. We assume that a user A chooses a random $x_A \in \mathbb{Z}_q$ and computes $y_A = g^{x_A} \mod p$. He keeps $x_A$ as his private key and publishes $y_A$ as his public key. In Schnorr’s signature scheme the signature of the user A on the message m is given by $(r_A, S_A)$, where,

$$r_A = h(g^{k_A} \mod m, m), \quad S_A = k_A - x_A \cdot r_A \mod p.$$

The signature are verified by checking the equality

$$r_A = h(g^{S_A} \cdot y_A \cdot r_A \mod p, m).$$
Threshold secret sharing scheme is a scheme to distribute a secret key K into n users in such a way that any t users can cooperate to reconstruct K but a collision of t – 1 or less users reveal nothing about the secret. There are many realization of this scheme, we shall use Shamir’s scheme. This scheme is based on Lagrange’s interpolation in a field. To implement it, a polynomial f of degree t – 1 is randomly chosen in Z_q such that

\[ f(0) = K. \]

Each user i is given a public identity \( u_i \) and a secret share \( f(u_i) \). Now any subset of t shareholders out of n shareholders can reconstruct the secret \( K = f(0) \), by pooling their shares and using

\[ f(0) = \sum_{i=1}^{t} f(u_i) \prod_{j=1,j \neq i}^{t} \frac{-u_j}{u_i - u_j} \mod q. \]

Here we assume that the authorized subset of t users consists of shareholders \( i \) for \( i = 1, 2, 3, \ldots t \).

3. THE PROPOSED SCHEME

The signer of the conventional digital signature schemes is usually a single person. But when the message is transmitted by an organization S to another organization R and may require the approval of more than one person then the responsibility of signing the messages needs to be shared by a subset \( H_S \) of t or more signer from a designated group \( G_S \) of n users belongs to the organization S. On the other hand, the signing group wants to generate the signature on a message \( m \) in such a way that the signature can be verified by any subset \( H_R \) of \( k \) or more users from a designated group \( G_R \) of \( l \) users belongs to the organization R, then threshold verification schemes serve our purpose. This paper proposes a threshold signature scheme with threshold verification.

We assume that the secret key of the organization S is \( x_S \) and the public key is \( y_S \) where \( y_S = g^{x_S} \mod p \) with \( x_S \in Z_q \). Similarly the organization R possesses a pair \((x_R, y_R)\) of private key \( x_R \) and public key \( y_R = g^{x_R} \mod p \). Also every user A in both the organizations possesses a pair \((x_A, y_A)\) with \( x_A \) secret and \( y_A = g^{x_A} \mod p \) public. We further assume that both the organization R and S have a common trusted center (CTC) for determining the group secret parameters of the two groups and also the secret shares all members. This scheme consists of the following steps:

3.1. GROUP SECRET KEY AND SECRET SHARES GENERATION FOR THE ORGANISATION S

(a). CTC selects the group public parameters \( p, q, g \) and a collision free one way hash function \( h \). CTC also selects for the group \( G_S \) a polynomial

\[ f_S(x) = a_0 + a_1x + \ldots + a_{t-1}x^{t-1} \mod q, \]

with \( a_0 = x_S = f_S(0) \).

(b). CTC computes the group public key, \( y_S \), as,

\[ y_S = g^{f_S(0)} \mod p. \]

(c). CTC randomly selects \( K \in Z_q \) and computes a public value \( W = g^K \mod p. \)

(d). CTC computes a public value \( v_{S_i} \) for each member of the group \( G_S \), as, \( v_{S_i} = f_S(u_{S_i}) \cdot y_S^k \mod p \). Here, \( y_S \) is public key and \( u_{S_i} \) is the public value associated with each user i in the group \( G_S \).

(e). CTC sends \{ \( v_{S_i}, W \) \} to each user i in the group \( G_S \) through a public channel.

3.2. GROUP SECRET KEY AND SECRET SHARES GENERATION FOR THE ORGANISATION R

(a). CTC selects for the group \( G_R \) a polynomial

\[ f_R(x) = b_0 + b_1x + \ldots + b_{k-1}x^{k-1} \mod q \]

with \( b_0 = x_R = f_R(0) \).

(b). CTC computes the group public key \( y_R = g^{f_R(0)} \mod p \).

(c). CTC computes a public value \( v_{R_i} \) for each member of the group \( G_R \), as, \( v_{R_i} = f_R(u_{R_i}) \cdot y_R^k \mod p \). Here, \( y_R \) is public key and \( u_{R_i} \) is the public value associated with each user i in the group \( G_R \).

(d). CTC sends \{ \( v_{R_i}, W \) \} to each user i in the group \( G_R \) through a public channel.

3.3. SIGNATURE GENERATION BT ANY \( t \) USERS

If a \( H_S \) is subset of \( t \) members of the organization \( S \) out of \( n \) members who agree to sign a message \( m \) to be sent to the organization \( R \), then the signature generation has the following steps:

(a). Each user \( i \in H_S \) randomly selects \( K_i, K_{i_2} \in Z_q \) and computes

\[ u_i = g^{-K_{i_2}} \mod p, v_i = g^{K_i} \mod p \]

and

\[ w_i = g^{K_{i_2} \cdot y_R^k} \mod p. \]

(b). Each user i broadcasts \( u_i, w_i \) publicly and \( v_i \) secretly to every other user in \( H_S \). Once all \( u_i, v_i \) and \( w_i \) are available, each member \( i, i \in H_S \) computes the product \( U_S, V_S, W_S \) and a hash value \( R_S \), as,

\[ U_S = \prod_{i \in H_S} u_i \mod q, V_S = \prod_{i \in H_S} v_i \mod q, \]

\[ W_S = \prod_{i \in H_S} w_i \mod q \]

and \( R_S = h(V_S, m) \mod q \).
(c). Each user $i \in H_S$ recovers his/her secret share $f_S(u_{S_i})$, as, $f_S(u_{S_i}) = v_{S_i}W^{x_{S_i}} \mod p$.

(d). Each user $i \in H_S$ modifies his/her shadow, as, $MS_{S_i} = f_S(u_{S_i}) \cdot \prod_{j=1, j \neq i}^{t} -u_{S_j} - u_{S_i} \mod q$.

(e). By using his/her modified shadow $MS_{S_i}$, each user $i \in H_S$ computes his/her partial signature $s_i$, as,

$$s_i = K_i + MS_{S_i} \cdot R_S \mod q.$$  

(f). Each user $i \in H_S$ sends his/her partial signature to the CTC, who produce a group signature $S_S = \sum_{i=1}^{t} s_i \mod q$.

(g). CTC sends $\{ S_S, U_S, W_S, m \}$ to the designated combiner $DC$ of organization $R$ as signature of the group $S$ for the message $m$.

3.4. SIGNATURE VERIFICATION BY THE ORGANISATION $R$

Any subset $H_R$ of $k$ users from a designated group $G_R$ can verify the signature. We assume that there is designated combiner $DC$ (can be any member among the members of the group $G_R$, or the head of the organization $R$), who collects partial computations from each user in $H_R$ and determines the validity of signature. The verifying process is as follows.

(a). Each user $i \in H_R$ recovers his/her secret share, as, $f_R(u_{R_i}) = v_{R_i}W^{x_{R_i}} \mod p$.

(b). Each user $i \in H_R$ modifies his/her shadow, as, $MS_{R_i} = f_R(u_{R_i}) \cdot \prod_{j=1, j \neq i}^{k} -u_{R_j} - u_{R_i} \mod q$.

(c). Each user $i \in H_R$ sends his modified shadow $MS_{R_i}$ to the $DC$. $DC$ computes $R_S$, as,

$$R_S = W_S \cdot U_S \mod q$$ 

and recovers $R_S = h \mod (R_S, m)$.

(d). $DC$ checks the following congruence $\gamma S_S \equiv R_S \cdot y_S^{R_S} \mod p$ for a valid signature.

If hold then $\{ S_S, U_S, W_S, m \}$ is a valid signature on the message $m$.

4. SECURITY DISCUSSIONS

In this sub-section, we shall discuss the security aspects of proposed Threshold Signature Scheme with threshold verification. Here we shall discuss several possible attacks. But none of these can successfully break our system.

(a). Can any one retrieve the organization’s secret keys $x_S = f_S(0)$ and $x_R = f_R(0)$, from the group public key $y_S$ and $y_R$ respectively?

This is as difficult as solving discrete logarithm problem. No one can get the secret key $x_S$ and $x_R$, since $f_S$ and $f_R$ are the randomly and secretly selected polynomials by the CTC. On the other hand, by using the public keys $y_S$ and $y_R$ no one also get the secret keys $x_S$ and $x_R$ because this is as difficult as solving discrete logarithm problem.

(b). Can one retrieve the secret shares, $f_S(u_{S_i})$ of members of $G_S$, from the equation

$$v_{S_i} = f_S(u_{S_i}) \cdot y_S^{K_i} \mod p.$$  

No because $f_S$ is a randomly and secretly selected polynomial and $K_i$ is also a randomly and secretly selected integer by the CTC. Similarly no one can get the secret shares $f_R(u_{R_i})$ of members of $G_R$, from the equation

$$v_{R_i} = f_R(u_{R_i}) \cdot y_R^{K_i} \mod p.$$  

(c). Can one retrieve the secret shares, $f_S(u_{S_i})$ of members of $G_S$, from the equation

$$s_i = f_S(u_{S_i}) \cdot W^{x_{S_i}} \mod p.$$  

Only the user $i$ can recover his secret shares, $s_i(u_{S_i})$, because $s_i$ is a randomly and secretly selected polynomial and $x_{S_i}$ is secret key of the user $i \in G_S$. Similarly no one can get the secret shares $f_R(u_{R_i})$ of members of $G_R$, from the equation

$$s_i = f_R(u_{R_i}) \cdot W^{x_{R_i}} \mod p.$$  

(d). Can one retrieve the modified shadow $MS_{S_i}$, integer $K_{i}$, the hash value $R_S$ and partial signature $s_i$, $i \in G_S$ from the equation

$$s_i = K_i + MS_{S_i} \cdot R_S \mod q.$$  

They all are secret parameters and it is computationally infeasible for a forger to collect the $MS_{S_i}$, integer $K_i$, the hash value $R_S$ and partial signature $s_i$, $i \in G_S$.

(e). Can the designated CTC retrieve any partial information from the equation,

$$S_S = \sum_{i=1}^{t} s_i \mod q.$$  

Obviously, it would be computationally infeasible for CTC to derive any information from $s_i$.

(f). Can one impersonate a member $i, i \in H_S$?

A forger may try to impersonate a shareholder $i, i \in H_S$, by randomly selecting two integers $K_i$ and $K_{i'} \in Z_q$ and

Copy Right © INDIACom – 2010
modifying $u_i$, $v_i$, and $w_i$. But without knowing the secret shares, $f_S(u_S)$, it is difficult to generate a valid partial signature $s_i$ to satisfy the verification equation,

$$S_S = \sum_{i=1}^{t} s_i \mod q \quad \text{and} \quad g^{S_S} = R_{S} \mod p.$$  

g. Can one forge a signature $\{S_S, U_S, W_S, m\}$ by the equation,

$$g^{S_S} = R_{S} \mod p?$$

A forger may randomly select an integer $R_{S}$ and then computes the hash value $R_{S}$ such that $R_{S} = h(R_{S}, m) \mod q$ bviously, to computes the integer $S_{S}$ is equivalent to solving the discrete logarithm problem. On the other hand, the forger can randomly select $R_{S}$ and $S_{S}$ first and then try to determine a value $R_{S}$, that satisfy the equation

$$g^{S_S} = R_{S} \mod p.$$  

However, according to the one-way property of the hash function $h$, it is quite impossible. Thus, this attack will not be successful.

(h). Can $t$ or more shareholders act in collusion to reconstruct the polynomial $f_S(x)$?

According to the equation

$$f_S(x) = \sum_{i=1}^{t} f_S(u_S) \prod_{j \neq i} \frac{x - u_S}{u_S - u_S} \mod q,$$  

the secret polynomial $f_S(x)$ can be reconstructed with the knowledge of any $t$ secret shares, $f_S(u_S)$, $i \in G_S$. So if in an organization the shareholders are known to each other, any $t$ of them may collude and find the secret polynomial $f_S$. This attack, however, does not weaken the security of our scheme in the sense that the number of users that have to collude in order to forge the signature is not smaller than the threshold value.

5. ILLUSTRATION

We now give an illustration in support of our scheme. Suppose $|G_S| = 7$, $|H_S| = 4$, $|G_R| = 6$ and $|H_S| = 5$. We take the parameters $p = 47$, $q = 23$ and $g = 2$.

5.1 GROUP SECRET KEY AND SECRET SHARES GENERATION FOR THE ORGANISATION $S$

(a). CTC selects for the group $G_S$ a polynomial $f_S(x) = 11 + 3x + 13x^2 + x^3 \mod 23$. Here $x_S = 11$ and $y_S = 2^{11} \mod 47 = 27$.

(b). CTC randomly selects $K = 8$ and computes $W = 2^8 \mod 47 = 9$.

(c). CTC computes the secret key, public key, secret share and public value for each member of the group $G_S$, as shown by the following table 1.

5.2 GROUP SECRET KEY AND SECRET SHARES GENERATION FOR THE ORGANISATION $R$

(a). CTC selects for the group $G_R$ a polynomial $f_R(x) = 7 + 2x + 4x^2 + 3x^3 \mod 23$. Here $x_R = 7$ and $y_R = 2^7 \mod 47 = 34$.

(b). CTC computes the secret key, public key, secret share and public value for each member of the group $G_R$, as shown by the table 2.

<table>
<thead>
<tr>
<th>USER</th>
<th>$y_{R_i}$</th>
<th>$x_{R_i}$</th>
<th>$u_{R_i}$</th>
<th>$f_R(u_{R_i})$</th>
<th>$v_{R_i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User – R1</td>
<td>9</td>
<td>15</td>
<td>11</td>
<td>21</td>
<td>14</td>
</tr>
<tr>
<td>User – R2</td>
<td>42</td>
<td>9</td>
<td>5</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>User – R3</td>
<td>27</td>
<td>11</td>
<td>8</td>
<td>21</td>
<td>16</td>
</tr>
<tr>
<td>User – R4</td>
<td>25</td>
<td>18</td>
<td>3</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>User – R5</td>
<td>17</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td>User – R6</td>
<td>14</td>
<td>13</td>
<td>4</td>
<td>18</td>
<td>32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VALUE</th>
<th>$y_{S_i}$</th>
<th>$x_{S_i}$</th>
<th>$u_{S_i}$</th>
<th>$f_S(u_{S_i})$</th>
<th>$v_{S_i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User – S1</td>
<td>7</td>
<td>12</td>
<td>2</td>
<td>8</td>
<td>34</td>
</tr>
<tr>
<td>User – S2</td>
<td>37</td>
<td>10</td>
<td>9</td>
<td>3</td>
<td>34</td>
</tr>
<tr>
<td>User – S3</td>
<td>28</td>
<td>14</td>
<td>8</td>
<td>22</td>
<td>38</td>
</tr>
<tr>
<td>User – S4</td>
<td>18</td>
<td>16</td>
<td>11</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>User – S5</td>
<td>8</td>
<td>21</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>User – S6</td>
<td>3</td>
<td>19</td>
<td>5</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>User – S7</td>
<td>36</td>
<td>17</td>
<td>4</td>
<td>19</td>
<td>40</td>
</tr>
</tbody>
</table>

5.3 SIGNATURE GENERATION

If $S_2$, $S_4$, $S_6$ and $S_7$ are agree to sign a message $m$ for the organization $R$, then

(a). $S_2$ selects $K_{x_1} = 5$, $K_{x_2} = 7$ and computes $u_2 = 18$, $v_2 = 32$ and $w_2 = 21$.

(b). $S_4$ selects $K_{x_4} = 4$, $K_{x_4} = 3$ and computes $u_4 = 6$, $v_4 = 16$ and $w_4 = 4$.

(c). $S_6$ selects $K_{x_6} = 12$, $K_{x_6} = 18$ and computes
A Directed Threshold Signature Scheme with Threshold Verification

\[ u_5 = 32, \ v_5 = 7 \text{ and } w_5 = 1. \]

(d) \( S_7 \) selects \( K_{i_1} = 21, K_{i_2} = 11 \) and computes \( u_7 = 7, v_7 = 12 \) and \( w_7 = 17. \)

(e) Each users computes \( U_S = 34, V_S = 3, W_S = 18 \) and \( R_S = h(3, m) = 8 \) (let).

(f) \( S_2 \) recovers \( f_S(u_{S2}) = 3 \) and computes \( MS_{S2} = 5 \) and \( s_2 = 22. \)

(g) \( S_4 \) recovers \( f_S(u_{S4}) = 4 \) and computes \( MS_{S4} = 21 \) and \( s_4 = 11. \)

(i) \( S_6 \) recovers \( f_S(u_{S6}) = 16 \) and computes \( MS_{S6} = 12 \) and \( s_6 = 16. \)

(j) \( S_7 \) recovers \( f_S(u_{S7}) = 19 \) and computes \( MS_{S7} = 19 \) and \( s_7 = 12. \)

(k) CTC produces a group signature \( S_2 = 15 \) and sends \{15, 34, 18, m\} to the designated combiner \( DC \) of organization \( R \) as signature of the group \( S \) for the message \( m. \)

5.4 SIGNATURE VERIFICATION

Suppose five users \( R_1, R_3, R_4, R_5 \) and \( R_6 \) want to verify the signature, then

(a) \( R_1 \) recovers \( f_R(u_{R1}) = 21 \) and computes \( MS_{R1} = 19. \)

(b) \( R_3 \) recovers \( f_R(u_{R3}) = 21 \) and computes \( MS_{R3} = 4. \)

(c) \( R_4 \) recovers \( f_R(u_{R4}) = 15 \) and computes \( MS_{R4} = 11. \)

(d) \( R_5 \) recovers \( f_R(u_{R5}) = 12 \) and computes \( MS_{R5} = 9. \)

(e) \( R_6 \) recovers \( f_R(u_{R6}) = 18 \) and computes \( MS_{R6} = 10. \)

(f) \( DC \) computes the value \( R_8 = 3 \) and then recovers \( R_8 = h(3, m) = 8. \)

(g) \( DC \) checks the following congruence for a valid signature

\[ 2^{15} \equiv 3 \cdot 27^8 \text{ mod } 45. \]

This congruence holds so \{15, 34, 18, m\} is a valid signature on the message \( m. \)

6. CONCLUSION

In this paper, we have proposed a **threshold signature scheme with threshold verification**. To obtain this construction, we have used the ElGamal public key cryptosystem and Schnorr's signature scheme. The security of this cryptosystem is based on the discrete log problem. The signature generation is done by certain designated sub-groups of the group of the receivers. Here designated sub-groups are characterized by threshold values. The threshold value can be different for signature generation and for signature verification. Until \( t \) (the threshold value of the group of senders) individuals act in collusion they will get no information about the group secret key. Similarly, until \( k \) (the threshold value of the group of recipients) individuals act in collusion they will get no information about the group secret key. The group public parameters \( p, q, g \) and a collision free one-way hash function \( h \) is same for both the organizations.

In any case of dispute between the group \( S \) and \( R \), the CTC keeps the records of signatures and plays the role of a trusted judge. Since the CTC can checks the validity of the signature so when any third party needs the signature verification, the CTC convince the third party about the facts.

**FUTURE SCOPE**

To produce/construct more efficient, secure and practical directed threshold signature schemes.

**REFERENCES**


Copy Right © INDIACom – 2010


