A Secure Designated Signature Scheme

Sattar J Aboud[1], Mohammed A Al-Fayoumi[1], Haidar S. Jabbar[2]
[1]Department of Computer Information Systems,
Faculty of IT, Middle East University for Graduate Studies (Jordan)

Annotation:

This paper presents a threshold designated receiver signature scheme that includes certain characteristic in which the signature can be verified by the assistance of the signature recipient only. The aim of the proposed signature scheme is to protect the privacy of the signature recipient. However, in many applications of such signatures, the signed document holds data which is sensitive to the recipient personally and in these applications usually a signer is a single entity but if the document is on behalf of the company the document may need more than one signer. Therefore, the threshold technique is employed to answer this problem. In addition, we introduce its use to shared signature scheme by threshold verification. The resultant scheme is efficient and dynamic.

Keywords*: Public key encryption, digital signature, designated signature, threshold signature scheme, threshold verification scheme.*

1- INTRODUCTION

Handwritten signature is a natural tool to authenticate the communication, but it is ineffective in e-messages. Therefore, an entity has to respond on another method such as digital signature. Digital signature is cryptography tool that solve this problem in e-communication. In essence digital signature has a characteristic that any entity have a copy of the signature can prove its validity by using certain public key, but no single one can forge the signature on another message. Such self authentication characteristic of digital signature is fairly suitable for many applications such as broadcasting of announcements and public key certificates, but it inappropriate for many other applications.

However, in many cases the signed document is sensitive to the signature recipient. For example, signature on medical records, tax document and other business transactions. For these cases the signature on the document must be verify and prove the validity by only the signature recipient to any trusted authority at any time required. Such signature is called designated receiver signature [1, 2]. In designated signature scheme, the receiver has an entire control over the signature verification operations; it means that no entity can check the validity of signature without recipient assistance. Certainly, the recipient must be capable to persuade any person that the signature is an authenticate signature signed by the signer. Such signature scheme is proposed to secure the confidentiality of the signature recipient by aiming to use the signature under the control of the recipient. The authorization ability is directly sent to the signature recipient in designated signature.

The idea of designated receiver signature scheme is initially introduced by Guillou and Quisquater in 1988 [3]. Another scheme introduced by Okamoto in 1994, which is more realistic structure of designated confirmer signature [4]. Also, Chaum in 1995 [5] presented the idea of designated confirmer signature to face the difficulty of vulnerability of undeniable signature that is the unavailability of authorization in the absence of the signer. The last two signature schemes are the same but just dissimilar in whom is given the ability of verifying the validity of the signature. In 2003, Hidenori Kuwakado and Hatsukazu Taaaka proposed another scheme using Trees [6]. Also, in 2006 Rongxing Lu and Zhenfu Cao suggest another scheme using RSA assumption [7]. Designated receiver signature can be utilized to design threshold cryptography [8] combined with secret sharing scheme.

In this paper we introduce a scheme of designated signature that is based on Shamir threshold signature scheme [9]. Also, we propose a method for apply threshold signature scheme, in which

the signature is verified just by assistance of an authorized group of designated verifiers. The proposed scheme is efficient and dynamic.

2- BACKGROUND

Suppose p and q are two large prime numbers where q divides $p-1$ and g is an integer where $g \equiv k^{(p-1)/q} \bmod p$, such that k is a random integer with $1 \le k \le p-1$ and $g > 1$, g is a generator of order q in z_p^*. Assume h is a one way hash function. The p, q, g, h are scheme keys and known to all entities. Suppose that each entity A in the scheme have a public and private key (e_A, d_A) respectively, where $e_A = g^{d_A} \bmod p$ such that $d_A \in Z_q$. The suggested designated receiver signature scheme relied on Schnorr signature scheme [10] and Shamir threshold scheme [9]. These two schemes are briefly explained.

In Schnorr signature scheme, entity A can produce a signature (a_A, b_A) on message m by finding $a_A = h\ (g^{r_A} \bmod p, m)$ using arbitrary $r_A \in Z_q$ and $b_A = r_A - d_A * a_A \bmod q$. This signature is verified by congruent $h(g^{b_A} * e^{a_A} \bmod p,\ m) \equiv a_A$

Secret sharing scheme allow a group of n participants to share a piece of secret information in which only authorized group of w participants can retrieve the secret. This scheme is named w out of n threshold scheme or simply (w, n) scheme. Shamir threshold secret sharing signature scheme is a method use to distribute a secret key R into n participants which is each group of w participants can participate to rebuild the secret R, but a collusion of $w-1$ or less participants disclose nothing about the secret.

Shamir threshold scheme is relied on Lagrange interpolation formula. To use it, a polynomial f of degree $w-1$ is randomly selected in Z_q where $f(0) = R$. Every participant i is provided a secret share $f(u_i)$ with u_i a public identity. Note that each group of w out of n participants can rebuild the secret $R = f(0)$. Thus $f(0) = \sum_{i=1}^{w} f(u_i) \prod_{j=1, j\neq i}^{w} \frac{-u_j}{u_i - u_j} \bmod q$, for simplicity we suppose that the authorized group of w participants includes shareholders i for $i = 1,2,..., w$

3- DESIGNATED RECIEVER SIGNATURE SCHEME

Assume that entity A needs to generate a signature for message m in which only entity B can verify the signature and also entity B can prove its validity to the trusted authority T when needed. The description of the scheme is as follows:

- **Algorithm for Signature Generation**

To generate the signature entity A must do the following:

1. Select randomly two integer numbers $r_{a_1}, r_{a_2} \in Z_q$
2. Finds $s_B = g^{r_{a_1} - r_{a_2}} \bmod p$
3. Finds $x_B = e_B^{r_{a_i}} \bmod p$
4. finds $a_A = h(x_B, s_B, m)$ using a hash function h
5. Computes $b_A = r_{a_2} - d_A * a_A \bmod q$
6. sends (s_B, a_A, b_A, m) to entity B, which is the signature when entity A on m

- **Algorithm for Signature Verification**

To verify the signature entity B must do the following:

1. Finds $c = (g^{b_A} * e_A^{a_A} * s_B) \bmod p$
2. Finds $x_B = c^{d_B} \bmod p$
3. Finds $h(x_B, s_B, m)$ and tests the validity of signature by $a_A = h(x_B, s_B, m) \bmod q$

- **Algorithm for Proof the signature Validity**

The steps of the algorithm that proof the validity of the signature by entity B to the trusted authority T are as follows:

1. Entity B sends (b_A, s_B, a_A, m, c) to trusted authority T
2. Entity T checks if $a_A = h(x_B, s_B, m) \bmod q$. If yes T stops the process; else go to the next steps.
3. Entity B in a zero knowledge protocol verifies to T that $\log_c x_B = \log_g * e_B$ using the discrete logarithm scheme in [2] which is as follows:
   - Entity T randomly selects $v, u \in z_p$ and finds $l = c^u * g^v \bmod p$, then send l to entity B.
   - Entity B randomly selects $D \in z_p$ and finds $Q = l * g^D \bmod p$
   - Finds $J = Q^{d_B} \bmod p$, and passes Q, J to entity T.
   - Entity T passes v, u to entity B, then entity B can verify that $l = c^u * g^v \bmod p$
   - Entity B sends D to entity T by which entity T can verify that $Q = c^u * g^{v+D} \bmod p$ and $J = x_B^u * e_T^{v+D} \bmod p$

4- THE PROPOSED THERSHOLD SIGNATURE SCHEME

Now, we introduce the proposed scheme for a designated receiver signature, which can employ the threshold sharing scheme. Assume that S is a set of n designated participants, out of which any w members can generate the signature on a message m for an entity B. The entity B can verify the signature and can prove its validity to any trusted authority T, when needed. Nobody can check the validity of the signature without the assistance of entity B. However, the description of the proposed threshold designated signature scheme is as follows:

Suppose an existence trusted authority T, which determine the group secrets keys and the secret shares v_i where $i \in S$. Assume M is any subset of S, including w members. Let also the existence of a designated combiner who collects partial signatures from every participant in the subgroup M. Each shareholder in the set has equal power with regard to the set secret. The generation of the designated signature requires w out of n shareholders and interaction with designated combiner. The steps for signature generation, signature verification and proof of validation of this scheme are as follows.

- **Algorithm for Set Secret Keys and Secret Shares Generation**

To generate the set secret keys and secret shares the trusted authority T must do the following:

1. Chooses the group public key p, q, g and a one way hash function h. Also, selects the polynomial $f(x) = a_0 + a_1 * x + ... a_{w-1} * x^{w-1} \bmod q$ such that $a_0 = R = f(0)$
2. Finds the set public key $e_S = g^{f(0)} \bmod p$
3. Finds a secret shares v_i for every member of the group S by compute $v_i = f(u_i) \bmod q$, where u_i is the public key associated with participant i in the group
4. Passes v_i to every participant in a secret way

- **Algorithm for Signature Generation**

When any w out of n members of the group agree to sign a message m for entity B, they generate the signature. Every member i must do the following:

1. Arbitrarily chooses R_{i_1} and $R_{i_2} \in z_q$
2. Finds $t_i = g^{R_{i_2} - R_{i_1}} \bmod p$
3. Finds $x_i = e_B^{R_{i2}} \bmod p$
4. Determines t_i public and x_i private available to every member of M. After t_i and x_i are available, each member finds X, W, K as follows:

1. $W = \prod_{i \in M} t_i \bmod p$
2. $X = \prod_{i \in M} x_i \bmod p$
3. $K = h(X, W, m) \bmod q$

5. Modified its share by $ms_i = v_i \prod_{j=1, j \neq i}^{w} \frac{-u_j}{u_i - u_j} \bmod q$
6. Uses its modifies share ms_i and arbitrary integer R_{i_1} to find the partial signature $s_i = R_{i_1} - ms_i * K \bmod q$
7. Passes its partial signature to the designated combiner who gathers the partial signatures and computes $V = \sum_{i=1}^{w} s_i \bmod q$
8. Trusted authority passes (V, W, K, m) to entity B as signature of the group S for the message m.

- **Algorithm for Signature Verification**

To verify the signature entity B must do the following:

1. Finds $c = g^V * (e_S)^K * W \bmod p$
2. Finds $X = c^{d_B} \bmod p$
3. Checks the validity of signature by $K = h(X, W, m) \bmod q$

- **Algorithm for validity Proof**

This part of the scheme is similar to the steps of the algorithm that proof the validity of the signature described in section 3. So we will not describe it in this section.

Example

The following example will support the proposed scheme for practical implementations. Suppose there are four participants. The four participants are $1,2,3$ and 4, any two participants say 1 and 4 can generate the designated signature on a message m for the entity B with public key $e_B = 8$ and private key $d_B = 6$. The algorithms of the example are as follows:

- **Algorithm for Set Secret Keys and Secret Shares Generation**

To generate the set secret keys and secret shares the trusted authority T must do the following:

1. Suppose T selects the group public key $p = 23, q = 11, g = 18$ and $f(x) = 3 + 5x \bmod 11$ such that $f(0) = 3$ which is the set private key.
2. Finds the group public value by $e_S = 18^3 \bmod 23 = 13$
3. Finds a secret shares v_i for every member of the group S by $v_i = f(u_i) \bmod q$, where u_i is the public key associated with participant i in the group as follows:

Participant	Public Key (u_i)		Secret Shares (v_i)
1	9	3 + 5 * 9 mod 11 =	4
2	12	3 + 5 * 12 mod 11 =	8
3	14	3 + 5 * 14 mod 11 =	7
4	16	3 + 5 * 16 mod 11 =	6

4. Passes v_i to every participant in a secret way

- **Algorithm for Signature Generation**

When any two participants say 1 and 4 out of four members of the group agree to sign a message m for an entity B, then the signature generation has the following steps:

1. The participant 1 arbitrarily chooses $R_{1_1} = 2$ and $R_{1_2} = 7$. Also, the participant 4 arbitrarily chooses $R_{4_1} = 5$ and $R_{4_2} = 9$
2. Then finds $t_1 = 18^{7-2} \bmod 23 = 3$ and $t_4 = 18^{9-5} \bmod 23 = 4$
3. Then finds $x_1 = 8^7 \bmod 23 = 12$ and $x_4 = 8^9 \bmod 23 = 9$
4. Both participants 1 and 4 determine (t_1, t_4) and (x_1, x_2) public by communication channel. As soon as (t_1, t_4) and (x_1, x_2) are available every member in M computes the result of X, W, K as follows:
   1. $W = 3*4 \bmod p = 12$
   2. $X = 12*9 \bmod p = 16$
   3. $K = h(16,12,m) \bmod q = 12*16 \bmod 11 = 5$
5. Participants 1 and 4 find their modified shares $ms_1 = 6$ and $ms_4 = 8$
6. Participant 1 uses it's modify share $ms_1 = 6$ and arbitrary integer $R_{i_1} = 2$ to find the partial signature $s_1 = 2 - 6*5 \bmod q = 2 - 30 = 2 - 35 = 33 \bmod 11 = 0 \therefore S_i = 5$. Also, Participant 4 uses it's modify share $ms_4 = 8$ and arbitrary integer $R_{4_1} = 5$ to find the partial signature $s_4 = 5 - 8*5 \bmod q = 5 - 40 = -35 = 33 \bmod 11 = 0 \therefore S_i = 9$
7. Both members pass the partial signature $(5,9)$ to the designated combiner who gathers the partial signatures by computes $V = \sum_{i=1}^{w} s_i \bmod q = 5 + 9 \bmod 11 = 3$
8. Trusted authority passes $(V = 3, W = 12, K = 5, m)$ to entity B as signature of the group S for the message m.

- **Algorithm for Signature Verification**

To verify the signature entity B must do the following:

1. Finds $c = 18^3 * 13^5 * 12 \bmod 23 = 3$
2. Finds $X = 3^6 \bmod 23 = 16$
3. Checks the validity of signature by $K = h(X = 16, W = 12, m) \bmod q = 5$

- **Algorithm for validity Proof**

This algorithm is to proof of validity by entity B to any trusted authority T. The steps of the algorithm are as follows:

1. Entity B sends $(V = 3, W = 12, K = 5, m, c = 3)$ to trusted authority T.
2. Entity T checks that $K = h(X = 16, W = 12, m) \bmod q = 5$. If this does not hold T stops the process; otherwise go to the next steps
3. Entity B proves to entity T that $\log_3 16 = \log_{18} 8$ in zero knowledge proof by using the following protocol
   1. Entity T selects randomly $u = 11, v = 13$ then finds $t = 2$ then passes t to entity B
   2. Entity B selects randomly $D = 17$ and finds $Q = 16$ and $J = 4$ then passes Q, J to entity T
   3. Entity T sends u, v to entity B, by which entity B can verify that $t = 2$
   4. Entity B passes D to entity T in which entity T can verify that $Q = 16$ and $J = 4$

Security of the Proposed Scheme

It is important to discuss the security of the proposed scheme from the following aspects:

1. It is impossible for any entity to recover the group secret key $f(0)$ from the group public key e_S since it is hard as finding the discrete logarithm problem.
2. It is impossible for any entity to recover the secret shares v_i from the public key u_i since f is a privately and randomly chosen polynomial.

3. It is impossible for designated combiner to recover the group secret key $f(0)$ or any partial information from the equation $V = \sum_{i=1}^{w} s_i \bmod q$ since is it computationally intractable.

5- Application of Registration Scheme

In this paper we introduce one of the most important applications of designated signature scheme which is the allocation of registration number.

Registration number is a widespread system in the world, such as the vehicle system, Internet shopping and other systems. However, in many situations it is essential to have a registration number for these systems. In this section we will suggest a registration scheme by which the registration number cannot be abused and forged. With this scheme the validity of an allocated registration number cannot verified at any time and by undetermined center. For the implementation of the proposed registration scheme we use a designated signature scheme.

The physical signature is employed for the allocation of registration number is followed by many official procedure and records. Unfortunately, the existing scheme is not highly secure. Suppose that an authority, offering the registration number for the public. Suppose also an entity A leads this authority and has a public and private key pair (e_A, d_A). Also, suppose a public entity B has a public and private key pair (e_B, d_B). Entity A creates a registration number with message m in order that entity B can directly collect the registration number. So, entity B can use the registration number publicly. Entity B is capable to verify its validity to any authorized participant C if needed. No individual other than entity B can use this registration number since only he can verify its validity. The steps of the proposed scheme are as follows:

Algorithm for allocation of registration number

To generate the allocation of registration number entity A must do the following:

1. Selects arbitrarily $r_{a_1}, r_{a_2} \in z_q$
2. Finds $s_A = g^{r_{a_1} - r_{a_2}} \bmod p$
3. Finds $x_B = e_B^{r_{a_1}} \bmod p$
4. Finds $a_A = h(x_B, s_A, m)$ using a hash function h
5. Computes $b_A = r_{a_2} - d_A * a_A \bmod q$
6. sends (s_A, a_A, b_A, m) to entity B, which is the signature when entity A on

Algorithm for collecting of registration number

To collect the registration number entity B must do the following:

1. Collects (s_A, a_A, b_A, m) and determine this public as the registration number
2. Finds $c = (g^{b_A} * e_A^{a_A} * s_A) \bmod p$
3. Finds $x_B = c^{d_B} \bmod p$
4. Verifies the validity of registration by calculating $a_A = h(x_B, s_A, m) \bmod p$

Algorithm for verification of registration number

This part is the same as in the previous section 3 algorithm for proof the signature validity. The steps of the algorithm that proof the validity of the signature by entity B to the trusted authority T are as follows:

1. Entity B passes (s_A, a_A, b_A, m, u) to entity T
2. Entity T checks if $a_A = h(x_B, s_A, m) \bmod q$. If this does not hold entity T stop the process; otherwise goes to the next steps.
3. Entity B in a zero knowledge protocol proves to entity T that $\log_c x_B = \log_g e_B$ as follows:

a. Entity T selects random $u, v \in z_p$ and computes $l = c^u * g^v \bmod p$ and send l to entity B
b. Entity B selects random $D \in z_p$ and computes $Q = l * g^D \bmod p$
c. Entity B computes $J = Q^{d_B} \bmod p$ and sends Q, J to entity T
d. Entity T sends u, v to entity B, by which entity B can verify that $l = c^u * g^v \bmod p$
e. Entity B sends D to entity T by which entity T can verify that $Q = c^u * g^{v+D} \bmod p$ and $J = x_B^u * e_T^{v+D} \bmod p$

Example

The following example supports the proposed scheme for practical implementation. Suppose $p = 23, q = 11, g = 5$. The public key and the private key of participants are as follows.

Participant	Private Key	Public Key
A	5	20
B	8	16

Algorithm for allocation of registration number

To generate the allocation of registration number entity A must do the following:

1. Suppose entity A selects $r_{a_1} = 7, r_{a_2} = 4$
2. Then computes $s_A = 5^{7-4} \bmod 23 = 10$
3. Then finds $x_B = e_B^{r_{a1}} \bmod p = 16^7 \bmod 23 = 18$
4. Then Finds $a_A = h(x_B = 18, s_A = 10, m = 1)$ suppose $m = 1 \therefore a_A = 18 * 10 * 1 \bmod 11 = 2$
5. Then finds $b_A = r_{a_2} - d_A * a_A \bmod q = 4 - 5 * 2 \bmod 11 = 5$
6. Sends $(s_A = 10, a_A = 2, , b_A = 5, m = 1)$ to entity B as the registration number

Algorithm for Collecting of registration number

To collect the registration number entity B must do the following:

1. Collects the registration number $(s_A = 10, a_A = 2, , b_A = 5, m = 1)$ and determine this public
2. Finds $c = (g^{b_A} * e_A^{a_A} * s_A) \bmod p = 5^5 * 20^2 * 10 \bmod 23 = 6$
3. Finds $x_B = c^{d_B} \bmod p = 6^8 \bmod 23 = 18$
4. Verifies the validity of registration by calculating $a_A = h(x_B = 18, s_A = 10, m = 1) \bmod q$

Algorithm for verification of registration number

The steps of the algorithm that proof the validity of the signature by entity B to the trusted authority T are as follows:

1. Entity B passes $(s_A = 10, a_A = 2, , b_A = 5, m = 1, u = 6)$ to entity T
2. Entity T checks if $a_A = h(x_B = 18, s_A = 10, m = 1) \bmod q$. If this does not hold entity T stop the process; otherwise goes to the next steps.
3. Now entity B in a zero knowledge protocol proves to entity T that $\log_6 18 = \log_5 16$. The description of this protocol is similar to the above.

Discussions

The above application services the allocation of registration number in the e-world with the following attributes

1. Only the participant can use his registration number as a result of the characteristic of designated signature scheme
2. The problems of forgery can be solved easily
3. By using this scheme we can minimize the probable misuse of the current scheme

4. The clear improvement of the scheme over current scheme is that the resulting registration number has no meaning to any third party
5. As the relation between the signature and the signer secret key is not known to anyone except to the designated receiver. Therefore, security level is greatly higher than any other scheme relied on discrete logarithm.

6- CONCLUSUION

The suggested designated receiver signature scheme can substitute the general digital signature schemes in various uses, especially if the signed document is sensitized to the recipient confidentiality. Besides the utilizing this signature scheme can also reducing the potential abuse in addition to the propagation of validating signature scheme. We introduced this signature scheme typed on the discrete logarithm intractability, employing a designated receiver signature and a secret sharing scheme. We also presented threshold verification of signature scheme in which is more secure and efficient in addition is a scalable and completely dynamic.

REFERENCES:

[1] Xun YI, Chik-How TAN and Eiji OKAMOTO, "Security of Kuwakado-Tanaka Transitive Signature Scheme for Directed Trees", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol.E87-A No.4 pp.955-957, 2004

[2] D. Chaum, "Zero-knowledge undeniable signatures", Advances in Cryptology-Eurocrypt '90, Springer-Verlag. LNCS 473, 1991, 458-461

[3] L. C. Guillou and J. J. Quisquater, "A Practical Zero-Knowledge Protocol Fitted to Security Micro-processors Minimizing both Transmission and Memory", Advances in Cryptology, Eurocrypt'88, Springer-Verlag, LNCS 330, 1988, 123-128.

[4] T. Okamoto, "Designated Confirmer Signatures and Public Key Encryption are Equivalent", Advances in Cryptology, Crypto'94, Springer-Verlag, LNCS 839, 1994, 61-74.

[5] D. Chaum, "Designated confirmer signatures", Advances in cryptology Eurocrypt '94, Springer Verlag, LNCS 950, 1995, 86-91

[6] Hidenori KUWAKADO and Hatsukazu TANAKA, "Transitive Signature Scheme for Directed Trees", IEICE\TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol.E86-A No.5 pp.1120-1126, 2003

[7] Rongxing Lu and Zhenfu Cao, "A Directed Signature Scheme Based on RSA Assumption", International Journal of Network Security, Vol.2, No.2, PP.153–157,. 2006

[8] Y. Desmedt and Y. Frankel, "Threshold Cryptosystems", Advances in Cryptology '89, Springer-Verlag, LNCS 435, 1990, 307-315.

[9] A. Shamir, "How to Share a Secret", Communication of ACM, 22(11), 1979, 612-613

[10] C. P. Schnorr, "Efficient Signature Generation by Smart Cards", Journal of Cryptology, 4(3), 1994, 161-174.

Article received: 2007-12-16
Received after processing: 2009-05-05