TECHNICAL NOTE

Threshold cryptosystem with multiple secret sharing policies

L. Harn
H.-Y. Lin
S. Yang

Indexing terms: Threshold cryptosystem, Secret sharing, Lagrange polynomial, ElGamal scheme

Abstract: In a group-oriented threshold cryptosystem, each group, instead of each individual member within the group, publishes a single group public key. An outsider can use this group public key to send encrypted messages to the group. However, the received encrypted messages can only be decrypted properly when the number of participating members is larger than or equal to the threshold value. All earlier solutions assume that there is only one secret sharing policy (i.e. one threshold value). We propose the first threshold cryptosystem with multiple secret sharing policies. In other words, a group can set up multiple secret sharing policies and a trusted key centre is responsible for selecting three publicly-known moduli, secret keys for group members, and publishing a corresponding public key for each policy during the initiation time. Moreover, there is only one single secret key kept by each group member. Whenever an outside wants to send a message to the group, he needs to determine how the message should be revealed to the group members, and therefore selects a proper public key to encrypt messages. Once the encrypted message is received by the group, according to the specified information, a predetermined number of group members is required to decipher the cipher-text. The ElGamal encryption scheme is used in the system with some modifications.

1 Introduction

The group-oriented threshold cryptosystem was first introduced by Desmedt [1] in 1987. In such a system, each group, instead of each individual group member, publishes a single public key. An outsider can use this group public key to send encrypted messages to the group. The received cipher-text can only be deciphered properly when the number of participating members is larger than or equal to a predetermined threshold value.

Paper 1003E (C3, E7), first received 4th March and in revised form 6th September 1993
The authors are with the Computer Science Telecommunications Program, University of Missouri — Kansas City, Kansas City, MO 64110, USA

Desmedt and Frankel proposed a solution based on the (t, n) threshold scheme and the ElGamal cryptosystem in Reference 2. But, to guarantee the solution works properly, it will have to perform the ElGamal system in $GF(2^m)$ and the Lagrange interpolation in Z_{2^m-1}, where $2^m - 1$ is a Mersenne prime. According to Reference 3, the computational difficulty of discrete logarithms in $GF(2^m)$ is not so hard as in the odd prime fields, $GF(p)$. Thus, m must be selected larger than 800 bits in order to achieve the proper security. Another solution with administrative clerks is discussed in Reference 4. However, it is not very robust because interactive communication between group members and the clerks is required. Two additional solutions without the assistance of a trusted party were proposed in References 5 and 6.

All earlier solutions to the threshold cryptosystem assume that there is only one secret sharing policy (i.e. one threshold value) for a group. It becomes more desirable for an outsider to send encrypted messages to a group which has several different secret sharing policies to reveal the encrypted message. To demonstrate the importance of this flexibility we use the following example.

Messages are addressed to a business company by using the threshold cryptosystem. The threshold value could depend on the content of the message which can be classified into different categories by the sender. For a message with crucial information of some business contracts, the cipher-text addressed to the company should be decrypted by more than one supervisor [1] (i.e. threshold value $\geqslant 2$). This arrangement can assure that this information is available to multiple supervisors simultaneously and so prevent one supervisor with the crucial information from leaving the company to start his own. Under emergency conditions, however, some urgent messages, such as 'send a rescue team to the factory', should be decrypted by any supervisor (i.e. threshold value = 1). This flexibility makes the system distinctive when compared with the single policy threshold cryptosystem.

In a straightforward approach we can implement the threshold cryptosystem with multiple secret sharing policies by using any existing threshold cryptosystem repeatedly for each threshold value and distributing multiple secrets to each group member. This approach, however, is very inefficient.

In this paper we propose the first efficient threshold cryptosystem with multiple secret sharing policies. In other words, a group can set up multiple secret sharing

policies and a trusted key centre is responsible for selecting secret keys for group members and publishing a corresponding public key for each policy during the initiation time. Moreover, in this scheme, there is only one single secret key kept by each group member. The scheme will perform the ElGamal system [7] and the Lagrange interpolation in $GF(p)$ and $GF(q)$, where p and q are two large odd primes.

2 Threshold cryptosystem with multiple secret sharing policies

The considered scheme allows a group to set up multiple secret sharing policies. It consists of four phases: the initiation phase, the policy setting up phase, the message encryption phase and the cipher-text decryption phase. Let us start with the initiation phase.

2.1 Initiation phase

Let n be the number of group members. A trusted key centre selects the following public parameters:

p = a large prime modulus, where $2^{511} < p < 2^{512}$
$w = (p-1)/2$, a large prime, where $2^{510} < w < 2^{511}$
q = a prime divisor of $w-1$, where $2^{159} < q < 2^{160}$

Then the key centre selects a distinct public value, $x_i \in [n+1, q-1]$, and distributes a secret value, $y_i \in [1, q-1]$, for each member i. Now, with n pairs of (x_i, y_i), a unique $(n-1)$th degree Lagrange interpolating polynomial, $f(x)$, can be determined as

$$f(x) = \sum_{i=1}^{n} y_i \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \bmod q \qquad (1)$$

where $f(x_i) = y_i$, $i = 1, 2, \ldots, n$. Later in the decryption phase, on the other hand, with the knowledge of any n pairs of (x_i, y_i), $f(x)$ can be uniquely reconstructed.

2.2 Secret sharing policy setting up phase

For a particular secret sharing policy with threshold value t, the key centre needs to determine a pair of secret and public keys. The key centre randomly selects integers, g and g', where $g = h^{(p-1)/w} = h^2 \bmod p > 1$ and $g' = h'^{(w-1)/q} \bmod w > 1$, with $0 < h < p$ and $0 < h' < w$. According to Fermat's theorem [8], for any generator g with order w, we have $g^r \bmod p = g^{r \bmod w} \bmod p$, where r is any nonnegative integer. Similarly, for any generator, g', with order q, we have $g'^r \bmod w = g'^{r \bmod q} \bmod w$. Thus, for any nonnegative integer r, we have $g^{g'^r} \bmod p = g^{g'^{r \bmod q} \bmod w} \bmod p$.

The secret key for this policy is determined by $g'^{f(0) \bmod q} \bmod w$, where $f(x)$ is the Lagrange polynomial determined by eqn. 1. The corresponding public key is determined by $g^{g'^{f(0)}} \bmod p$. Since later in the decryption phase, only t ($t < n$) members are required to determine the decryption key, which is a function of the secret key $g'^{f(0)} \bmod w$ (i.e. not to reveal this value directly), $n - t$ additional public shadows are required to help these t members to derive the decryption key. We denote these public shadows as: $g'^{f(1)}, g'^{f(2)}, \ldots, g'^{f(n-t)}$. Just like the group public key, $g^{g'^{f(0)}}$, all these public shadows need to be determined by the key centre during this phase and all these keys are known only to all group members.

The key centre will make $(t, g, g^{g'^{f(0)}})$ the group's public key known to all outsiders and $(g'^{f(1)}, g'^{f(2)}, \ldots, g'^{f(n-t)})$ the corresponding public shadows known to all internal members. Similarly, for different sharing policies with different threshold values, the key centre works in the same way to determine the corresponding group's secret key, public key and public shadows. Note that the generator g', used to determine the secret and public keys for each policy, should be different although the secret key used by each member remains unchanged.

2.3 Message encryption phase

We incoporate the ElGamal encryption scheme [7] to encrypt the message. Suppose an outsider A wants to send one secret message m to the group and decides to choose secret sharing policy with threshold value t. First, A accesses the group public-key directory and obtains group public key, $(t, g, g^{g'^{f(0)}})$. Then A randomly selects a number r from $[0, w-1]$. According to the public key distribution scheme proposed by Diffie and Hellman [9] in 1976, a common secret key K, which will be shared by A and the group, can be obtained by A as

$$K = (g^{g'^{f(0)}})^r \bmod p$$

Then A computes

$$C_1 = g^r \bmod p$$

$$C_2 = K \times m \bmod p$$

The cipher-text blocks $\{C_1, C_2\}$ is transmitted to the group.

2.4 Cipher-text decryption phase

Once the group receives $\{C_1, C_2\}$ from A, any t members need to be connected in any order to generate the common secret key K first. Without losing generality, we assumed the members are connected sequentially in the following order: $u_1, u_2, \ldots, u_t$, and use their secret keys to generate the common secret key K on a progressive basis. With the knowledge of the public shadows $(g'^{f(1)}, g'^{f(2)}, \ldots, g'^{f(n-t)})$ and the secret key $f(x_1) = y_1$, u_1 gets C_1 and computes

$$SK_1 = (C_1)^{\left(\prod_{v=1}^{n-t} g'^{f(v)} \prod_{j=1, j\neq v}^{n-t} \frac{-j}{v-j} \prod_{j=1}^{t} \frac{-x_j}{v-x_j} \bmod q\right)\left(g'^{f(x_1)} \prod_{j=1}^{n-t} \frac{-j}{x_1-j} \prod_{j=1, j\neq 1}^{t} \frac{-x_j}{x_1-x_j} \bmod q\right) \bmod w} \bmod p$$

SK_1 is sent to u_1's successor u_2. u_2 uses his secret key $f(x_2) = y_2$ to compute

$$SK_2 = (SK_1)^{\left(g'^{f(x_2)} \prod_{j=1}^{n-t} \frac{-j}{x_2-j} \prod_{j=1, j\neq 2}^{t} \frac{-x_j}{x_2-x_j} \bmod q\right) \bmod w} \bmod p$$

Just by repeating the same procedure until the tth member has used its secret key to work on the value obtained from its predecessor, the common secret key K can be finally obtained as $K = SK_t$.

Theorem 1: $SK_t = K$

Proof

We can rewrite SK_t as

$$SK_t = (C_1)^{\left(\prod_{v=1}^{n-t} g^{f(v)\prod_{j=1,j\neq v}^{n-t}\frac{-j}{v-j}\prod_{j=1}^{t}\frac{-x_j}{v-x_j} \bmod q}\right)\left(\prod_{v=1}^{t} g^{f(x_v)\prod_{j=1}^{n-t}\frac{-j}{x_v-j}\prod_{j=1,j\neq v}^{t}\frac{-x_j}{x_v-x_j} \bmod q}\right) \bmod w} \bmod p$$

$$= (g^r)^{\left(g^{\left(\sum_{v=1}^{n-t} f(v)\prod_{j=1,j\neq v}^{n-t}\frac{-j}{v-j}\prod_{j=1}^{t}\frac{-x_j}{v-x_j}\right) \bmod q}\right)\left(g^{\left(\sum_{v=1}^{t} f(x_v)\prod_{j=1}^{n-t}\frac{-j}{x_v-j}\prod_{j=1,j\neq v}^{t}\frac{-x_j}{x_v-x_j}\right) \bmod q}\right) \bmod w} \bmod p$$

$$= (g^r)^{g^{\left(\sum_{v=1}^{n-t} f(v)\prod_{j=1,j\neq v}^{n-t}\frac{-j}{v-j}\prod_{j=1}^{t}\frac{-x_j}{v-x_j}+\sum_{v=1}^{t} f(x_v)\prod_{j=1}^{n-t}\frac{-j}{x_v-j}\prod_{j=1,j\neq v}^{t}\frac{-x_j}{x_v-x_j}\right) \bmod q} \bmod w} \bmod p$$

$$= (g^r)^{g^{f(0) \bmod q} \bmod w} \bmod p$$

$$= K$$

QED

Once the common key K has been obtained, the corresponding message m is decrypted as

$$m = K^{-1} \times C_2 \bmod p$$

where K^{-1} is the multiplicative inverse of K and p.

3 Security and discussion

3.1 Security analysis

Several possible attacks are proposed here. For each attack, the difficulty of breaking our system is reliant on breaking the discrete logarithm problem.

(*a*) The difficulty of deriving the group's secret key from the corresponding group's public key and the public shadows of each secret sharing policy is equivalent to solving the discrete logarithm problem.

(*b*) During the deciphering phase, although all t members use their secret keys to compute the common secret key K, the security of their secret keys are still preserved due to the discrete logarithm problem.

(*c*) The secret key K, shared by the outsider and the group, will differ from one message to the next, if the value of r is different. Hence r should be selected randomly by the outsider to insure the scheme's security.

(*d*) If the corresponding generators g'_1, g'_2 and g'_3, for three different policies are $g'_1 \times g'_2 = g'_3 \bmod p$, the secret key, $g'^{f(0)}_3$, can be determined by just multiplying the two other secret keys, $g'^{f(0)}_1$ and $g'^{f(0)}_2$. Thus, we suggest selecting a different prime for each generator, g'_i to avoid this attack.

3.2 Discussion

(*a*) A single key is required by every group member to derive the corresponding secret keys for all different secret sharing policies.

(*b*) For each policy, there are public keys, $(t, g, g^{g'f(0)})$, known to all outsiders, and public shadows, $(g'^{f(1)}, g'^{f(2)}, \ldots, g'^{f(n-t)})$, known to all internal members. Thus, the storage for the public values is in fact increased. In addition, every time a key is decrypted, an exponentiation has to be performed for each public shadow, that might be some overhead for this proposed scheme. But, the advantage of having only a single secret key per user might outweigh these disadvantages, especially when $n - t$ is small.

(*c*) Our scheme performs the secret sharing and encryption in $GF(q)$ and $GF(p)$, where q and p are two large odd primes.

4 Conclusion

A group-oriented threshold cryptosystem with multiple secret sharing policies is proposed. A trusted key centre is responsible for selecting three publicly-known moduli, secret keys for group members and publishing a corresponding public key for each policy during the initiation time. Each member keeps only a single secret key and the group has multiple public keys for outsiders to encrypt their messages addressed to the group.

5 References

1 DESMEDT, Y.: 'Society and group oriented cryptography: a new concept', *Adv. Cryptology, Proc. Crypto '87*, Santa Barabara, USA, 16th–20th Aug. 1988, pp. 120–127

2 DESMEDT, Y., and FRANKEL, Y.: 'Threshold cryptosystem', *Adv. Cryptology, Proc. Crypto '89*, Santa Barbara, USA, 20th–24th Aug. 1989, pp. 307–315

3 ODLYZKO, A.M.: 'Discrete logs in a finite field and their cryptographic significance', *Adv. Cryptology, Proc. Eurocrypt '84*, Paris, France, Apr. 1984, pp. 224–314

4 FRANKEL, Y.: 'A practical protocol for large group oriented networks', *Adv. Cryptology, Proc. Eurocrypt '89*, Belgium, 10th–13th Apr. 1989, pp. 56–61

5 PEDERSEN, T.P.: 'A threshold cryptosystem without a trusted party', *Adv. Cryptology, Proc. Eurocrypt '91*, Brighton, UK, 8th–11th Apr. 1991, pp. 522–526

6 LAIH, C.S., and HARN, L.: 'Generalized threshold cryptosystems', *Adv. Cryptology, Proc. Asiacrypt '91*, Japan, 11th–14th Nov. 1991, pp. 159–169

7 ELGAMAL, T.: 'A public key cryptosystem and a signature scheme based on discrete logarithms', *IEEE Trans. Inform. Theory*, July 1985, **IT-31**, pp. 469–472

8 'The digital signature standard', *Comm. ACM 35*, 1992, (7), pp. 36–40

9 DIFFIE, W., and HELLMAN, M.E.: 'New directions in cryptography', *IEEE Trans. Inform. Theory*, Nov. 1976, **IT-22**, pp. 644–654