TECHNICAL NOTE

Efficient sharing (broadcasting) of multiple secrets

L. Harn

Indexing terms: Secret sharing, Threshold scheme, Discrete logarithm

Abstract: Instead of using the conventional m out of n perfect secret sharing scheme to protect a single secret among n users, the authors propose a secret sharing scheme based on one cryptographic assumption to protect multiple secrets. It is shown that, with this relaxation of the security requirement, secret sharing and some related secret-sharing problems, such as cheater detection and secret broadcasting, can be solved very efficiently.

1 Introduction

As described by Denning [1], an (m, n) threshold scheme is designed to break the single master key K up into n different 'shadows' $K_1, K_2, \ldots, K_n$ such that:

(*a*) with the knowledge of any m shadows ($m \leqslant n$), the master key K can be easily derived; and

(*b*) with the knowledge of any $m-1$ or fewer shadows, it is impossible to derive the master key K.

Shamir and Blakley independently published the first two threshold schemes based on the Lagrange interpolating polynomial and the linear projective geometry, respectively, in 1979 [2, 3]. Since then, several schemes have been proposed [4, 5].

Although many (m, n) linear-threshold schemes provide Shannon's perfect secrecy up to the threshold value m, unfortunately they require a large data expansion (i.e. m shadows are needed to reclaim one secret). Therefore, these methods are very inefficient as a conveyor of information [6]. In 1984, Blakely [6] presented the idea of a (d, m, n) ramp scheme, in which partial information will be obtained if at least d, but no more than m, shadows are known. This scheme provides a more efficient way of sharing information. In Reference 7, a so-called 'dynamic threshold scheme' has been developed. Although the reconstructed secret changes dynamically, the threshold value decreases in proportion to the number of different secrets which have been revealed. One secret-sharing scheme with the ability of sharing up to l secrets is proposed by Harn and Lin [8]; it deals with generalised secret-sharing policies, based on the RSA [9] assumption.

In this note, we propose a new concept of secret sharing among n users. Instead of providing Shannon's perfect secrecy, as in most secret sharing schemes, the security of our proposed scheme is based on the discrete-logarithm problem. We assume that there are multiple secrets shared among a group of users and that revealing each secret requires the cooperation of a different number of users. Thus, we can set up a threshold value ($m = 2$, or 3, or ..., or n) for each secret according to its security requirement. Our scheme will provide just a single shadow for each user, and with the knowledge of any m (with $1 \leqslant m \leqslant n$) shadows a unique secret can be easily derived. In addition, we will show that, with the relaxation of security requirement from the perfect secrecy to one cryptographic assumption, our scheme can also handle some secret-sharing problems, such as cheater detection and secret broadcasting, very efficiently.

Paper 1874E (C3, C14), first received 22nd July 1994 and in final revised form 18th January 1995

The author is with the Computer Science Telecommunications Program, University of Missouri-Kansas City, Kansas City, MO 64110, USA

2 Scheme for sharing multiple secrets

Our scheme utilises the cryptographic techniques of Shamir's perfect secret-sharing scheme based on the Lagrange interpolating polynomial and the digital-signature algorithm proposed by NIST [10]. It allows a group of n users to share multiple secrets and, with the knowledge of at least m shadows, a unique secret can be determined. Without losing generality, we first assume that there are k different secrets with the same threshold value m.

The dealer selects:

p = a large prime modulus, where $2^{799} < p < 2^{800}$;
q = a prime divisor of $p-1$, where $2^{159} < q < 2^{160}$ and $q^2 \nmid (p-1)$;
$\{a_i$, for $i = 0, \ldots, m-1\}$, and $f(x) = a_0 + a_1 x + \cdots + a_{m-1}x^{m-1} \bmod q$, each a_i is a random integer with $0 < a_i < q$;
$\{g_i$, for $i = 0, \ldots, k\}$, where $g_i = h_i^{(p-1)/q} \bmod p$, each h_i is a random integer with $0 < h_i < p$ such that $h_i^{(p-1)/q} \bmod p > 1$. Each g_i is a generator with order q in GF(p).

$\{a_i, i = 0, \ldots, m-1\}$ are the secret values and $\{p, q, g_i\}$ are the public values. Note that, according to lemma 1 in Reference 10, if g_i is a generator with order q in $GF(p)$, we have $g_i^t \bmod p = g_i^{t \bmod q} \bmod p$, for any nonnegative integer t.

For each secret S_i there is a corresponding public value g_i. The secrets are determined by

$$S_i = g_i^{a_0} \bmod p$$
$$= g_i^{f(0)} \bmod p \quad \text{for } i = 1, 2, \ldots, k$$

Shadow generation:
The shadow assigned to each shareholder is computed as

$$k_i = f(x_i) \bmod q \quad \text{for } i = 1, 2, \ldots, n$$

where $x_i > m$ is the public value associated with shareholder i. The dealer also needs to compute one public key y_i as

$$y_i = g_0^{k_i} \bmod p$$

for shareholder i and a public value y_0 as

$$y_0 = g_0^{a_0} \bmod p$$

for verification purposes. We will discuss the functions of these parameters in Section 4.

Secret reconstruction:
To reconstruct the secret S_i, each shareholder j needs to use his secret shadow k_j to compute the subshadow for secret S_i as

$$k_{i,j} = g_i^{k_j} \bmod p$$

Suppose that we have m shareholders, whose public values are $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$, and subshadows are $k_{i,i_1}, k_{i,i_2}, \ldots, k_{i,i_m}$, a secret S_i' can be reconstructed as

$$S_i' = k_{i,i_1}^{w(1)} \quad k_{i,i_2}^{w(2)} \quad \cdots \quad k_{i,i_m}^{w(m)} \bmod p$$

where

$$w(u) = \prod_{j=1, j\neq u}^{m} \frac{(-x_{i_j})}{(x_{i_u} - x_{i_j})} \bmod q$$

Theorem 1: S_i is equivalent to the secret S_i.

Proof: With m shadows $k_{i_1}, k_{i_2}, \ldots, k_{i_m}$, $f(0)$ can be reconstructed from the Lagrange polynomial as

$$f(0) = \sum_{s=1}^{m} k_{i_s} w(s) \bmod q$$

We have

$$\begin{aligned} S_i &= g_i^{a_0} \bmod p \\ &= g_i^{f(0)} \bmod p \\ &= g_i^{\sum_{s=1}^{m} k_{is} w(s) \bmod q} \bmod p \\ &= g_i^{k_{i1} w(1) \bmod q} \quad g_i^{k_{i2} w(2) \bmod q} \quad \cdots \quad g_i^{k_{i3} w(m) \bmod q} \bmod p \\ &= k_{i,i_1}^{w(1)} \quad k_{i,i_2}^{w(2)} \quad \cdots \quad k_{i,i_m}^{w(m)} \bmod p \\ &= S_i' \end{aligned}$$

Discussion
The security of the above scheme is based on the difficulty of solving the discrete-logarithm problem. Although the subshadow $k_{i,j}$ is revealed to others while deriving the secret S_i, the private shadow k_j for user j is still kept secret. This is due to the fact that, since $k_{i,j} = g_i^{k_j} \bmod p$, with the knowledge of $k_{i,j}$, g_i, to solve k_j is equivalent to solving the discrete-logarithm problem. Similarly, knowing any secret S_i will not harm the security of any other secret S_l. This is because each secret S_i is generated by a different generator g_i randomly selected by the dealer. One possible attack is that, if the attacker knows some relations among these generators g_i, for $i = 0, 1, \ldots, k$, the attacker can derive some unrevealed secrets with the knowledge of some revealed secrets. For example, if the attacker knows that $S_1 = g_1^{a_0} \bmod p$, $S_2 = g_2^{a_0} \bmod p$, and there exists g_3 such that $g_3 = g_1 g_2 \bmod p$, the secret S_3 can be easily derived by $S_3 = S_1 S_2 \bmod p$. In our scheme, since each g_i is randomly selected by the dealer (not by the attacker) from the set which contains 2^{160} integers, the probability of having g_3 to satisfy $g_3 = g_1 g_2 \bmod p$ is 2^{-160}, which is extremely low and can be almost neglected. The difficulty of finding other possible relations among generators is equivalent to the difficulty of exhausting the space which contains 2^{160} integers, and is also extremely hard, especially when the number of generators is much less than the size of the space (i.e. $k \ll 2^{160}$). For some applications, in which the submitted subshadows and the secret reconstructed are revealed to the public each time, our scheme allows each shareholder to use the same private shadow repeatedly to generate different subshadows and derive different secrets. The security of all these secrets is never compromised.

Let us consider another concept of secret sharing among n users. We assume that there are n secrets and that each secret has a distinct security requirement. Therefore, we can set up a threshold value m (i.e. $m = 1, 2, \ldots, n$) for each secret according to its security requirement. Just as in the previous scheme, the dealer selects public parameters p, q and $\{g_i, \text{ for } i = 0, 1, \ldots, n\}$ and secret parameters $\{a_i, \text{ for } i = 0, 1, \ldots, n-1\}$. Then the private shadow assigned to each shareholder is computed as

$$\begin{aligned} k_i &= f(x_i) \bmod q \\ &= a_0 + a_1 x_i + \cdots + a_{n-1} x_i^{n-1} \bmod q \\ &\qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

where x_i is the public value associated with shareholder i and $x_i \notin \{1, 2, \ldots, n-1\}$. Assume that S_i is the secret with threshold value i. For each secret S_i, there is a corresponding public value g_i. The secret is determined by

$$S_i = g_i^{a_0} \bmod p \quad \text{for } i = 1, 2, \ldots, n$$

To reconstruct the secret S_i, at least i subshadows are required. The subshadow for shareholder j is computed as

$$\begin{aligned} k_{i,j} &= g_i^{f(x_j)} \bmod p \\ &= g_i^{k_j} \bmod p \end{aligned}$$

Since $f(x)$ is an $(n-1)$th-degree polynomial, in total n subshadows are required to reveal the secret. In addition to the i subshadows presented from the shareholdes, it requires $n - i$ additional public subshadows to obtain S_i. The dealer can publish these $n - i$ public subshadows as

$$g_i^{f(1)} \bmod p \quad g_i^{f(2)} \bmod p \quad \cdots \quad g_i^{f(n-i)} \bmod p$$

Incorporating these $n - i$ public subshadows with at least i subshadows from the shareholders, the secret S_i can therefore be determined in the same way as the previous one.

Note that, for secrets with smaller threshold values, there are more public subshadows associated with them.

Although we have published $n(n-1)/2$ public subshadows, these public subshadows cannot compromise the secrets.

3 Detection of cheaters

The verifiable secret-sharing scheme is to provide the shareholder with the abilities to verify that (*a*) the secret shadows obtained from the dealer are derived consistently from the same secret; and (*b*) the secret shadows obtained from the other shareholder in the secret-reconstruction process are genuine shadows. These abilities are very important. For example, a dishonest dealer can cheat some shareholders by giving them fake shadows. Communication errors (i.e. noise) can also result in fake shadows. A shareholder may also cheat the others in the secret-reconstruction process by presenting

a fake shadow to prevent others from obtaining the real secret. Interested readers can refer to Reference 11 for up-to-date research on this subject. Some solutions, such as those described in References 12–17, to prevent shareholder from cheating others, are unconditionally secure. On the other hand, some solutions, such as those in References 18 and 19, which prevent the dealer from cheating shareholders, rely on computational assumptions regarding certain encryption schemes.

Here, we would like to show that our proposed scheme can prevent either the dealer or the shareholder from cheating others.

Detection of cheater: dealer is the prover
In our scheme, the dealer needs to publish one public key y_i as

$$y_i = g_0^{k_i} \bmod p$$

for each shareholder and one public value y_0 as

$$y_0 = g_0^{a_0} \bmod p$$

for the verification purpose.

Each shareholder, with the knowledge of his own secret shadow k_i and all these public values, is able to verify his secret shadow.

Theorem 2: Any received fake shadow from the dealer can be easily detected by any shareholder.

Proof: Let us first examine the situation of fake shadows caused by communication noise. Shareholder i receives a fake shadow k_i' and the corresponding public key is $y_i = g_0^{k_i} \bmod p$. Obviously, this fake shadow can be easily detected by the shareholder.

On the other hand, let us examine what will happen if an honest dealer picks up all the real shadows consistently. From theorem 1, for any m real public keys, y_{i_j}, $j = 1, 2, \ldots, m$, we have

$$y_0 = y_{i_1}^{w(1)} \quad y_{i_2}^{w(2)} \quad \cdots \quad y_{i_m}^{w(m)} \bmod p$$

where

$$w(u) = \prod_{j=1, j\neq u}^{m} \frac{(-x_{i_j})}{(x_{i_u} - x_{i_j})} \bmod q$$

Since any combination of m out of n the public values satisfies the above relation, each shareholder can verify that his secret shadow is derived consistently by the dealer. It is worth noting that the detection procedure has exponential effort, since C_m^n is exactly $\binom{n}{m}$.

Detection of cheaters: shareholder is the prover
Suppose that a shareholder j is a prover and wants to prove the validity of his subshadow $k_{i,j} = g_i^{k_j} \bmod p$ to a verifier. They can execute the following interactive protocol.

Verifier: Randomly selects an integer v within $[1, q-1]$ and computes

$$C = (g_0 g_i)^v \bmod p$$

C is sent to the prover j.

Prover: With his secret shadow k_j the prover computes

$$C' = C^{k_j} \bmod p$$

C' is sent back to the verifier.

Verifier: With prover's public key y_j, the verifier examines whether the following equation holds:

$$C'^{v-1} = y_j \times k_{i,j} \bmod p$$

If this equation does hold, the subshadow is accepted. Otherwise, it is a fake one.

Theorem 3: If $C'^{v-1} = y_j \times k_{i,j} \bmod p$, the subshadow $k_{i,j}$ is a genuine subshadow.

Proof: This proof is straightforward.

4 Broadcasting multiple secrets

Ref. 20 describes a scheme to incorporate the threshold scheme to allow a conference chairman to broadcast a conference key to all conference participants. Since the conference key itself is a secret to a group of users, the same approach has been proposed to allow a transmitter to broadcast a secret to a group of intended listeners [21].

Cryptanalysis of the scheme proposed in References 20 and 21
The scheme in References 20 and 21 can be described as follows. Assume that the transmitter (conference chairman) A has already shared a common secret key K_{Ai} with each intended listener with identity ID_i, where ID_i is the public information associated with each intended listener. Now the transmitter wants to broadcast a secret S to t intended listeners, ID_i, for $i = 1, 2, \ldots, t$. The transmitter can design a $(t+1, n)$ threshold scheme based on the Lagrange interpolating polynomial $f(x)$ with polynomial values $f(ID_i) = K_{Ai}$, for $i = 1, 2, \ldots, t$, and $f(0) = S$. In other words, with any $t+1$ shadows it is possible to derive S. The transmitter broadcasts t additional shadows, which are polynomial values $f(x_i)$, for $i = 1, 2, \ldots, t$, where $x_i \notin \{0, ID_i | \text{ for } i = 1, 2, \ldots, t\}$. Every intended listener, with the knowledge of these received t shadows and the one additional shadow, which is its own common secret key shared with the transmitter, can derive the secret S. An intruder, who can only intercept t public shadows, cannot obtain enough information to derive S. We use the following example to illustrate this scheme.

Example 1: Assume that A is the transmitter and shares a common secret key K_{AB} with B, and a common secret key K_{AC} with C. A wants to broadcast a secret S_1 to both B and C. Then A can design a $(3, n)$ threshold scheme based on the second-order Lagrange interpolating polynomial $h_1(x) = a_{2,1}x^2 + a_{1,1}x + S_1 \bmod p$, with $h_1(ID_B) = K_{AB}$, $h_1(ID_C) = K_{AC}$ and $h_1(0) = S_1$, and p is a prime integer larger than the secret S_1. Then A needs to compute and broadcast two additional shadows, $h_1(1)$ and $h_1(2)$. With the knowledge of these two public shadows and one private shadow, either K_{AB} or K_{AC}, B and C are able to reconstruct this polynomial $h_1(x)$ and derive the secret $S_1 = h_1(0)$.

Security discussion
The above scheme is insecure. Since B and C are able to reconstruct the polynomial $h_1(x)$, and ID_B, ID_C are public information, B is able to find the common secret key $h_1(ID_C) = K_{AC}$ shared between C and A, and C is also able to find the common secret key $h_1(ID_B) = K_{AB}$ shared between B and A.

One possible solution is to include the ID_i as each user's secret to avoid this attack. However, from the following example, we can see that this modified scheme is still insecure for broadcasting multiple secrets.

Example 2: We continue on our previous example. Assume that A wants to broadcast two secrets S_1 and S_2 to both B and C. Two second-order Lagrange interpolating polynomials $h_1(x) = a_{2,1}x^2 + a_{1,1}x + S_1 \bmod p$ and $h_2(x) = a_{2,2}x^2 + a_{1,2}x + S_2 \bmod p$ are generated. After deriving two secrets S_1 and S_2, B also knows two polynomials $h_1(x)$ and $h_2(x)$. Thus B can establish two equations as

$$h_1(ID_C) = a_{2,1}ID_C^2 + a_{1,1}ID_C + S_1 \bmod p$$

and

$$h_2(ID_C) = a_{2,2}ID_C^2 + a_{1,2}ID_C + S_2 \bmod p$$

where $h_1(ID_C)$, $h_2(ID_C)$ and ID_C are C's secrets. Since $h_1(ID_C) = h_2(ID_C) = K_{AC}$, by combining above two equations, B is able to solve C's secret ID_C. Using the similar approach, C is able to solve B's secret ID_B.

Broadcasting multiple secrets

The conference chairman becomes the dealer, as described in Section 2, to select a secret conference key and to distribute this key to all participating users. The conference chairman selects p, q and $\{g_i, \text{for } i = 0, \ldots, k\}$ as described in Section 2. We use the following example to explain our scheme.

Example 3: Example 1 is used again here. Assume that A is the transmitter and shares a common secret key K_{AB} with B and a common secret key K_{AC} with C. A wants to broadcast a secret $S_1 = g^{k_1} \bmod p$ to both B and C, where k_1 is a random integer. Then A can design a $(3, n)$-threshold scheme based on the second-order Lagrange interpolating polynomial $h_1(x) = a_{2,1}x^2 + a_{1,1}x + k_1 \bmod p$, with $h_1(ID_B) = K_{AB}$, $h_1(ID_C) = K_{AC}$ and $h_1(0) = k_1$. Then A needs to compute and broadcast two additional shadows $g^{h_1(1)} \bmod p$ and $g^{h_1(2)} \bmod p$. With the knowledge of these two public shadows and one private shadow, either $g^{K_{AB}}$ or $g^{K_{AC}}$, B and C are able to derive the secret S_1 according to the following procedure. B uses his own secret shadow $g^{h_1(ID_B)} = g^{K_{AB}} \bmod p$ to compute

$$\begin{aligned} & g^{K_{AB}[(-1)-2)/(ID_B-1)(ID_B-2)] \bmod q} \; g^{h_1(1)[(-ID_B)(-2)/(1-ID_B)(1-2)] \bmod q} \; g^{h_1(2)[(-ID_B)(-1)/(2-ID_B)(2-1)] \bmod q} \bmod p \\ &= g^{K_{AB}[(-1)(-2)/(ID_B-1)(ID_B-2)] + h_1(1)[(-ID_B)(-2)/(1-ID_B)(1-2)] + h_1(2)[(-ID_B)(-1)/(2-ID_B)(2-1)] \bmod q} \bmod p \\ &= g^{h_1(0)} \bmod p \\ &= S_1 \end{aligned}$$

Similarly, C can use his own secret shadow $g^{h_1(ID_C)} = g^{K_{AC}} \bmod p$ to obtain S_1.

Discussion

The security of this proposed scheme is the same as is discussed in Section 2 and is based on the difficulty of solving the discrete-logarithm problem.

This proposed scheme differs slightly from the scheme previously cryptanalyzed. In that scheme, any arbitrary value can be selected as the secret. In the modified scheme, however, the secret is computed indirectly from a one-way function and it cannot be pre-determined by the transmitter.

5 Conclusion

We have shown a method of sharing multiple secrets based on one cryptographic assumption. Our scheme will provide a single shadow for each user, and, with the knowledge of any t shadows, where t is a variable (i.e. $1 \leqslant t \leqslant n$), a unique secret can be easily derived. We also demonstrate its flexibility to handle cheaters detecting, and broadcasting multiple secrets.

6 References

1 DENNING, D.E.R.: 'Cryptography and data security' (Addison–Wesley, 1982), pp. 179–185
2 SHAMIR, A.: 'How to share a secret', *Commun. ACM*, 1979, **22**, pp. 612–613
3 BLAKLEY, G.R.: 'Safeguarding cryptographic keys'. Proceedings of NCC, 1979, AFIPS Press, Montvale, NJ, vol. 48, pp. 313–317
4 DAVIDA, G.I., DEMILLO, R.A., and LIPTON, R.J.: 'Protecting shared cryptographic keys'. Proceedings of Symposium on Security and Privacy, IEEE Computer Society Press, 1980, pp. 100–102
5 KOTHARI, S.C.: 'Generalized linear threshold scheme'. Proceedings of CRYPTO '84, Springer Verlag, Berlin, 1984, pp. 231–241
6 BLAKLEY, G.R., and MEDDOWS, C.: 'Security of ramp schemes'. Proceedings of CRYPTO '84, Springer Verlag, Berlin, 1984, pp. 242–268
7 LAIH, C.S., HARN, L., and LEE, J.Y.: 'Dynamic threshold scheme based on the definition of cross-product in an N-dimensional linear space'. Proceedings of CRYPTO '89, Springer Verlag, Berlin, 1989, pp. 271–277
8 HARN, L., and LIN, H.Y.: 'An 1-span generalized secret sharing scheme'. Proceedings of CRYPTO '92, Springer Verlag, Berlin, 1992, pp. 558–565
9 RIVEST, R.L., SHAMIR, A., and ADELMAN, L.: 'A method for obtaining digital signatures and public-key cryptosystem', *Comm. ACM*, 1978, **21**, (2), pp. 120–126
10 The Digital Signature Standard, *Comm. ACM*, 1992, **35**, (7), pp. 36–40
11 BRICKELL, E.F.: 'The detection of cheaters in threshold schemes'. Proceedings of CRYPTO '88, Springer Verlag, Berlin, 1988, pp. 564–577
12 McELIECE, R.J., and SARWATE, D.V.: 'On sharing secrets and Reed-Soloman codes', *Comm. ACM*, 1981 **24**, pp. 583–584
13 TOMPA, M., and WOLL, H.: 'How to share a secret with cheaters', *J. Cryptology*, 1988, **1**, (2), pp. 133–138
14 SIMMONS, G.: 'An introduction to shared secret schemes and their applications'. Sandia report SAND88-2298, 1988
15 RABIN, T., and BEN-OR, M.: 'Verifiable secret sharing and multiparty protocols with honest majority'. Proceedings of the 21st Annual ACM Symposium on the Theory of Computing, 1989, pp. 73–85
16 FELDMAN, P.: 'A practical scheme for non-interactive verifiable secret sharing'. Proceedings of the 28th FOCS, IEEE, 1987, pp. 427–437
17 PEDERSEN, T.P.: 'Non-interactive and information-theoretic verifiable secret sharing'. Proceedings of CRYPTO '91, Springer Verlag, Berlin, 1992, pp. 129–140
18 CHOR, B., GOLDWASSER, S., MICALI, S., and AWERBUCH, B.: 'Verifiable secret sharing and achieving simultaneity in the presence of faults'. Proceedings of 26th IEEE Symposium on Foundations of the Computer Science, 1985, pp. 372–382
19 BENALOH, J.C.: 'Secret sharing homomorphisms: keeping shares of a secret secret'. Proceedings of CRYPTO '86, Springer Verlag, Berlin, 1986, pp. 251–260
20 LAIH, C.S., HARN, L., and LEE, J.Y.: 'A new threshold scheme and its application in designing the conference key distribution cryptosystem', *Inf. Process. Lett.*, 1989, **32**, (3), pp. 95–99
21 BERKOVITS, S.: 'How to broadcast a secret'. Proceedings of EUROCRYPT '91, Springer Verlag, Berlin, 1991, pp. 535–541