Secret Sharing

See:
Shamir, *How to Share a Secret*, CACM, Vol. 22, No. 11, November 1979, pp. 612–613

How to Keep a Secret Key Securely

Information can be secured by encryption under a secret key.
The ciphertext can be replicated. Even if one replicated copy is lost, or stolen, the information remains available and secure.
Therefore, the problem of securing information reduces to the problem of securing the secret key.

- Encrypting the key does not help - need to secure another key
- Replicating the key itself is insecure

Goal: Distribute the key to a group, without revealing the key to any subgroups

Application

- An employee looses a key ⇒ The company looses the encrypted information
- In particular: Quits his job, dies
- If a copy of the key (or information) is given to others to protect against lost, they can use the key (or information)
- In general, we need a way to share a secret in a group, without revealing the secret to any party (or small subgroup), and without allowing any party to inhibit reconstruction

Secret Sharing

Definition: Secret Sharing schemes

- Sharing a secret between n parties
- Each party receives a share
- Cooperation of predefined subgroups enables to reconstruct the secret
- Smaller subgroups cannot reconstruct the secret, nor any information on the secret

(k, n)-Threshold Schemes

Definition: (k, n)-Threshold Schemes satisfy

- Sharing a secret between n parties
- Each party receives a share
- Cooperation of any k parties enables to reconstruct the secret
- Single parties, or subgroups of up to $k-1$ parties, cannot reconstruct the secret, nor any information on the secret

Example

Let S be a 56-bit DES key.
We can share it between two parties by giving each party 28 bit of S.

- By cooperation they can recover the full key.
- However, each party gets 28 bits of information on the key.
- If S was used to encrypt a file, each party can search only 2^{28} keys without cooperation, rather than 2^{56}.

This is not a valid threshold scheme.

A $(2,2)$-Threshold Scheme

- Let S be an m-bit secret.
- Choose an m-bit uniformly random value, S_1.
- Compute $S_2 = S \oplus S_1$.
- The two shares are S_1 and S_2.
- Give S_1 to Alice, and S_2 to Bob.

A $(2,2)$-Threshold Scheme (cont.)

Theorem:
Security: Neither Alice nor Bob receive any information on S when they get S_1 or S_2.
Proof:
Alice: S_1 is uniformly random and independent of the secret.
Bob: Given S_2, there is 1–1 relationship between candidates for S_1 and S, thus, $P(S=k) = P(S_1 = k \oplus S_2) = 2^{-m}$, for any S, and therefore $H(S) = H(S_1) = m \Rightarrow$ no information leak. QED
Theorem:
Correctness: Alice and Bob can collaborate and recover the secret S.
Proof:
$S = S_1 \oplus S_2$. QED

An (n, n)-Threshold Scheme

We can extend the previous scheme to an (n, n)-threshold scheme

- Let S be an m-bit secret.
- Chose $n-1$ m-bit uniformly random $S_1, S_2, \ldots, S_{n-1}$.
- Compute $S_n = S \oplus S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1}$.
- The n Shares are $S_1, S_2, \ldots, S_n$.

Relation to One-Time-Pad

One time pads (perfect ciphers) can be used as $(2,2)$-threshold schemes, by choosing S_1 randomly, and computing $S_2 = D_{S_1}(S)$.
On the other hand, $(2,2)$-threshold schemes can be used as one time pads, where S is the plaintext, S_1 is the key, and S_2 is the ciphertext.

Shamir's (k, n)-Threshold Schemes

These schemes are based on unique interpolation of polynomials:

- Given k points on the plane $(x_1, y_1), \ldots, (x_k, y_k)$, where all the x_i's are distinct, there exists an unique polynomial of degree $k-1$ for which $q(x_i) = y_i$ for all i

Shamir's (k, n)-Threshold Schemes (cont.)

The Scheme:

- Let S be a secret $S \in \mathcal{S}$.
- Select a prime modulus p, $p > \max(n, |\mathcal{S}|)$.
- Select a random polynomial $q(x)$ such that $q(0) = S$, i.e., select the coefficients $a_1, a_2, \ldots a_{k-1} \in Z_p$ randomly, and select $a_0 = S$.
- The Polynomial is

$$q(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} \pmod{p}$$

- Distribute the shares

$$\begin{aligned} S_1 &= (1, q(1)) \\ S_2 &= (2, q(2)) \\ &\vdots \\ S_n &= (n, q(n)) \end{aligned}$$

Shamir's (k, n)-Threshold Schemes (cont.)

Theorem:
The secret S can be reconstructed from every subset of k shares
Proof:
q is a polynomial of degree $k-1$, thus given k points it can be uniquely reconstructed.
By Lagrange, given k points (x_i, y_i), $\quad i = 1, \ldots, k$

$$q(x) = \sum_{i=1}^{k} y_i \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{x - x_j}{x_i - x_j}$$

and in our case

$$S = q(0) = \sum_{i=1}^{k} y_i \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{-x_j}{x_i - x_j} \pmod{p}$$

QED

Shamir's (k, n)-Threshold Schemes (cont.)

Theorem:
Any subset of up to $k-1$ shares does not leak any information on the secret
Proof:
Given $k-1$ shares (x_i, y_i), every candidate secret S corresponds to an unique polynomial of degree $k-1$ for which $q(0) = S$. From the construction of the polynomials, all their probabilities are equal. Thus, $H(S)$ remains $\log |\mathcal{S}|$.
Conclusion: Secret sharing is perfectly secure, and does not depend on the computation power of any party.

Remarks

1. Lagrange interpolation require $O(k^2)$ computation steps. Efficient computations can be performed in $O(k \log^2 k)$.

2. When S is long, we can divide it to shorter blocks and share each block.

3. The size of each share is the same as the size of the secret.

4. It is possible to add new shares (i.e., increasing n), whenever required, without affecting the other shares.

5. It is possible to remove shares without affecting the other shares (as long as the share is really destroyed).

6. It is easy to replace all the shares, or even k, without changing the secret, and without revealing any information of the secret, by selecting a new polynomial $q(x)$, and a new set of shares.

7. It is possible to give some parties more than one share. For example, in a company:

Remarks (cont.)

- The president: 3 shares
- Each vice president: 2 shares
- Each director: 1 share

A $(3, n)$-threshold scheme allows the

- president, or
- two vice presidents
- vice president and a director
- any three directors

to recover the key (sign checks, open the safe, etc.).

Some Extensions

There are several extensions to secret sharing schemes:

1. Visual secret sharing (Naor, Shamir).
2. Supporting more complex access structures.
3. Verifiable secret sharing.
4. Sharing a computation (secure-multiparty computation).