Secrect Sharing Progress

Joan Ngure

Report

Perfect Secrecy:
An algorithm (share,reconstruct) is perfectly secure if given the secret space M, the share space S and threshold t then:

$$Pr(M|S_1 \cdots S_{t-1}) = Pr(M)$$

In shamir's secret sharing scheme for example, if participants less that t collude then the most powerful set will be t-1 they have fewer unknowns than t-x for t $>$ x $>$ 1.
Consider the following example:

$$\begin{gathered}(\text{n,n}) \\ \text{Secret} = 18630967 \\ S_1 = 18 \\ S_2 = 63 \\ S_3 = 09 \\ S_4 = 67\end{gathered}$$

If the first three participants collude, then they will learn more information about the secret than they initially new. In this example:

$$Pr(M|S_1 \cdots S_{t-1}) \neq Pr(M)$$

Thus the example satisfy correctness but not secrecy.

Cheater detection:
In shamir's secret sharing scheme a cheater can go undetected. Consider the following example:

$$\begin{gathered}\text{secret} = 6 \in \mathbb{Z}_7 \\ \text{scheme (3,5)} \\ P_1(1,5), P_2(2,5), P_3(3,6), P_4(4,1), P_5(5,4)\end{gathered}$$

If first three participants decide to reconstruct the secret and participant one gives an incorrect share say 4. The reconstructed secret is as follows:

$$4 \times \tfrac{5\times4}{6\times5} + 5 \times \tfrac{6\times4}{1\times6} + 6 \times \tfrac{6\times5}{2\times1} = 3$$

$3 \in \mathbb{Z}_7$ - legal
$3 \neq 6$ - incorrect.
In this example, s=p where p is the prime modulo operation and s is an integer less than p such that the secret $\in \{0, 1 \cdots s-1\}$.
One may think that choosing s≠p may increase the probability of cheater detection. The following example shows that this is not the case. In fact a single participant can deceive t-1 participants and obtain the secret.
Suppose the first three participants decide to pool their shares and participant one decides to cheat. The participant:

1. Constructs a polynomial $f'(x)$ using interpolation such that:
   $f'(0) = -1$ and $f'(P_2) = \cdots f'(P_t) = 0$
2. Participant one announces their share + $f'(P_1)$
3. The reconstructed polynomial is $f(x)+f'(x)$ where $f(0)+f'(0) =$ secret-1

If the secret is not 0 then the deception will go undetected.
Using the above example:
$f'(x) = -1 \times \frac{(x-2)\times(x-3)}{(0-2)\times(0-3)} = 2$, for x = 1
Participant one will produce 2+5 =0. Thus:

$$0 \times \frac{5\times4}{6\times5} + 5 \times \frac{6\times4}{1\times6} + 6 \times \frac{6\times5}{2\times1} = 5$$

secret-1 = 5
secret = 6
Participant one successfully deceives t-1 participants.
One straight forward way to prevent cheating is when the dealer assigns a share with an unforgeable signature along with it. This however, depends on the hardness of integer factorization hypothesis or secure encryption schemes and would require an additional signature scheme along with it, which has its own complications. Tompa and Woll proposed an improvement on shamir's secret sharing scheme.

1. Choose a prime p.
2. Choose randomly $a_1, \cdots, a_{t-1} \in \mathbb{Z}_p$
3. Form $f(x) = D + \sum_{i=1}^{t-1} a_i x^i$ mod p
4. Choose $(x_l, x_z, \cdots, x_n)$ uniformly and randomly from among all permutations of n distinct elements from $\{1, 2, \cdots, p-1\}$. Let $D_i = (x_i, d_i)$, where $d_i = f(x_i)$.

The only difference is step 4. Instead of have such a single polynomial we have several where one of them gives the secret.
If t-1 participants conspire then they will form functions $f(x_{i_1}), f(x_{i_2}), \cdots, f(x_{i_{t-1}})$

which are uniformly distributed and mutually independence hence the secret shares $D_{i_1}, D_{i_1}, \cdots, D_{i_{t-1}}$ reveal no information about the secret D.
If t-1 participants decide to deceive participant t and present fabricated shares assuming the know f(x) hence they know the secret, the will only succeed if $f_{D'}(x_{i_t}) = f(x_{i_t})$ and $D' \neq D$. The probability of deceiving participant i_t is at most:

$$\frac{(s-t)(t-1)}{(p-t)} < \epsilon.$$

Although cheaters are detected with a high probability, they still succeed in obtaining the secret while other participants don't. A simple way to solve this is adding a dummy variable say s to the set of all possible secret $\{1, 2, \cdots, s-1\}$ which is never used as the value of the secret. The secret is now encoded as a sequence of $D^1, D^2, \cdots, D^t$ where $D^i = D$ for some random i and $D^j = s$ for $i \neq j$. When t participants pool their shares they reconstruct the secret then $D^1, D^2, \cdots$ until $D^j \neq s$ is obtained which terminates the protocol. If D^j is not legal then cheating has occurred.