Secrect Sharing Progress

Joan Ngure

Secret Sharing Made Short

Cheating and Cheater Detection

Recall:
Distribution Scheme:

1. Choose a random encryption key K. Encrypt the secret S using the encryption function ENC under the key K, let $E = ENC_x(S)$.
2. Using Information dispersal algorithm (IDA) partition the encrypted file E into n fragments, $E_1, E_2, \cdots, E_n$
3. Using perfect secret sharing scheme (PSS) generate n shares for the key K, denoted $K_1, K_2, \cdots, K_n$.
4. Send to each participant $P_i, i = 1, 2, \cdots, n$ the share $S_i = (E_i, K_i)$. The portion K_i is privately transmitted to P_i (e.g. using encryption or any other secure way).

Reconstruction Scheme:

1. Collect from m participants $P_{ij}, j = 1, 2, ., ., m$ their shares $S_i, = (E_{ij}, K_{ij})$.
2. Using IDA reconstruct E out of the collected values $E_{ij}, j = 1, 2, \cdots, m$.
3. Using PSS recover the key K out of $K_{ij}, j = 1, 2, \cdots, m$.
4. Decrypt E using K to recover the secret S.

This scheme assumes that all participants return correct shares, hence there is no way to detect a cheater [2]. A cheater can either return a modified share of the secret, key or both and go undetected. (my argument) Since the secret is shared using IDA, a cheater who wishes to obtain the secret and deceive the other participants into obtaining the wrong secret, would submit an incorrect share of the key. Why? If a cheater submits an incorrect share of the secret, there is no scheme that shows how to reconstruct the secret after. It is also hard to tell which part of the fragment was incorrectly reconstructed. The cheater therefore has a higher chance of obtaining the secret if an incorrect share of the key is and shamir's secret sharing scheme is used (as suggested in the paper).

In [3] we saw how a cheater can succeed in obtaining the secret and deceiving other participants into obtaining the incorrect secret using secret sharing scheme, where in our case it is the key. The cheater:

1. Constructs a polynomial $f'(x)$ using interpolation such that:
   $f'(0) = -1$ and $f'(P_2) = \cdots f'(P_m) = 0$ where P_i for $i = 1, 2 \cdots m$ are the shares of the key for m participants.
2. The cheater announces their share + $f'(P_1)$ assuming P_1 is their key share.
3. The reconstructed polynomial is $f(x) + f'(x)$ where $f(0) + f'(0) =$ key - 1

If the key is not 0 then the deception will go undetected. [3] 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 key.
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 key shares $D_{i_1}, D_{i_1}, \cdots, D_{i_{t-1}}$ reveal no information about the key D.
If t-1 participants decide to deceive participant t and present fabricated shares assuming the know f(x) hence they know the key, they 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 key 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 key. The key 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 key 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.

Cheating detection in secret shared using IDA: One of the solutions that was proposed was ensuring the shares are signed using the private signing key of the dealer. The solution is space efficient but has the following drawbacks:

1. Requires implementation of public key system to support public signatures where the computation, key management and administration is costly.
2. Identity of the dealer should be known.
3. To avoid replay attacks, a time-stamp mechanism is needed.

The solution given in this paper that caters for the above mentioned drawbacks was proposed in [1] **distributed fingerprints**. The scheme uses a one-way hash function H such that it is infeasible for a polynomial-time adversary to find a collision for instance given strings x and y such that H(x) = H(y). The scheme also uses an error correcting code. It has two functions namely; coding and decoding denoted by C and D respectively. The coding function C, maps a string of length k into a sequence of n (number of participants) strings while the decoding function D reconstructs the string from a sequence of n strings as long as there are at least $\frac{d-1}{2}$ correct strings where d is the distance of the code.

How does distributed finger print work?

1. From our scheme each share distributed to the participants is denoted by E_i.
2. Fingerprint each E_i for $i = 1, \cdots, n$ as follows:
   (a) Compute H(E_i)
   (b) Compute using the coding function C(H(E_i)) $= e_i, e_2, \cdots, e_n$.
   (c) Distribute the corresponding share e_i to each party i for $i = 1, \cdots, n$

Reconstruction:

1. Collect from each person their share E_i and the n fingerprint shares given to them.
2. Use the decoding function to reconstruct D$(e_i, e_2, \cdots, e_n)$ = H(E_i) and H(E_i'), where E_i' is the share submitted during reconstruction.
3. If H(E_i) = H(E_i') accept E_i' as the correct information E_i otherwise reject. Reconstruct the file using the correct shares.

This scheme can be simplified, by concatenating all the hash values $H(F_i)$ and applying the coding algorithm on this string rather than doing it to each $H(F_i)$ independently. Each participant therefore gets the pair(E_i, e_i). At the time of reconstruction each participant submits their pair where the decoding function D is used to reconstruct the string. The correct E_is are determined and used to recover the file.
[2] also suggests distributed fingerprints as a solution to the key cheating too. Cheating is therefore detected with a non-negligible probability.