Secrect Sharing Progress

Joan Ngure

On the share efficiency of robust secret sharing and secret sharing with cheating detection [2]

In a (t,n) threshold SSS the adversary is passive.
Security goal- Unauthorized subsets do not learn any information about the secret.
Consider corrupted parties that submit incorrect shares and there are extra security goals with respect to incorrect shares.

Requirements:

1. The secret can be recovered up to t incorrect submitted shares (robust secret sharing).
2. Cheating detection here is to prevent another player from reconstructing an invalid secret.

0.1 Robust Secret Sharing

Satisfy's the property that a secret can be reconstructed even if some players provide incorrect shares.

Adversarial Capability

1. Unbounded adversary
2. Corrupts upto t players
3. If a player P_i is corrupted, the adversary learns their share and controls the information the player sends to reconstruct(R).

We have **rushing** and **non-rushing** adversaries. Rushing decides what the corrupted players send to R depending on what they have seen so far while non-rushing selects the corrupted shares before the start of each round.

Privacy- No t players have any information about the secret. Take a subset of at most size t, call it B.

$$\forall s_1, s_2 \in S$$
$$Pr[\text{secret is } s_1|\text{view}_B] = Pr[\text{secret is } s_2|\text{view}_B]$$

Where view_B is the total information members of B see and S is the set of possible secrets.

Robust game

1. Share distribution phase: Dealer computes shares and gives them to the participants.
2. Reconstruction phase: Adversary corrupts upto t players.
3. Final phase: R has all n shares with at most t corrupts ones and outputs s′ based on the shares. The adversary wins if $s' \neq s$.
   Adversary advantage: $Adv^{\text{Robust}}_{\pi,(t,n)}(A) = Pr[s' \neq s]$

A (t,n) threshold robust secret sharing scheme Π(share,rec) is said to be unconditionally secure with (t,δ)- robust property against non-rushing adversary if it is both perfect privacy and $Adv^{\text{Robust}}_{\pi,(t,n)}(A) \leq \delta$ in the above game. δ denotes error probability in reconstructing the correct secret.

0.2 Basic RSSS

[1] has an approach you have your secret s and select a random value r (s,r $\in \mathbb{Z}_p$). Evaluate p = s·r. Shares of s and r are distributed privately but p is public. Knowledge of p and t shares does not reveal anything about the secret. During reconstruct, the relation p = s·r must hold for s to be accepted.This approach however does not guarantee the required robustness. From this direction, we use [3] information checking vectors and construct an efficient RSS with unconditional security.

The Information Checking Protocol
This process enables us to carry out authentication of information. It is not as powerful as digital signatures, but it does not require cryptographic assumptions on the other hand.
The dealer (D) who has the secret s, an intermediary (INT) who receives s from D and the recipient R who receives s from INT. R accepts s, if they believe it originated from D. The protocol of acceptance is wished to have the following properties:

1. If D and R are honest then R will accept s if it actually originated with D, and will reject with probability $\geq 1 - \frac{1}{2^k}$ any value of s$'$. k is the security parameter.

2. Whether D is honest or dishonest, INT will know with probability $\geq \frac{1}{2^k}$ whether R will accept s that he holds

For this, information checking tool is proposed. It has two major parts:

1. Check vectors

2. Verification of check vectors

Information checking is carried out by the three participants in the following way:
We assume that a large prime p> 2^k, has been decided upon for all computations. s$\in \mathbb{Z}_p$.

Check vectors
D chooses two random numbers $b \neq 0, y \in \mathbb{Z}_p$ and gives INT the pair (s,y). D computes s+by=c and hands R vector (b,c) which is known as check vector.
INT later transmits (s,y) to R and R computes s+by and accepts iff it equals c. The check vector reduces the probability that INT will send a false s' which R will accept.

Lemma 0.1. *The probability that INT will deceive R, when D is honest is* $\frac{1}{p-1} < \frac{1}{2^k}$

Proof. If INT chooses a new value s' which they would like R to accept, then they must find y' that solves the equation

$$s' + y' = c$$

Only on y' will solve the equation for the b held by R. Thus the probability

$$\Pr(\text{R will accept } s' \neq s | \text{ D is honest}) = \frac{1}{p-1}$$

□

Lemma 0.2. *R has no information about s vfrom their check vector.*

Proof. All values of s are possible with equal probability and for each s there is a single y which satisfies the equation. □

Verification of Check Vectors Zero- knowledge proof technique is used. The procedure is transferred as follows: D sends INT ordered set of pairs $(s, y_1 \cdots s, y_{2k})$ and R $(b_1, c_1) \cdots (b_{2k}, c_{2k})$ and

$$s + b_i y_i = c_i \text{ for } 1 \leq i \leq 2k$$

INT chooses k distinct points $d_1, \cdots, d_k$ for $1 \leq i \leq 2k$. R is then asked by INT to reveal $(b_{d_1}, c_{d_1}) \cdots (b_{d_k}, c_{d_k})$. For each of these check vectors INT computes:

$$s + b_{d_i} y_{d_i} = c_{d_i}$$

If all k check vectors satisfy the equation, INT concludes that R will accept the value s, otherwise R will reject this value.

Lemma 0.3. *The probability that the intermediary INT will assume that R will accept s when in fact R will reject it is at most* $\leq \frac{1}{\binom{2k}{k}}$

Proof. If the dealer is a knight then R will never reject s. In order for the error stated in the Lemma to occur, it must be that INT has received k check vectors which are good, and R holds k unrevealed check vectors which are all faulty. Thus, the probability that INT will choose all the k good check vectors is: Pr(INT assumes R will accept s | R rejects s) $\leq \binom{2k}{k}^{-1}$ □

Thus using the protocol of Information Checking as a primitive, we achieve the required 1 and 2.

0.3 The scheme:

A group of n players. Let t and n be positive integers and n = 2t+1.
Share algorithm : D chooses random r,X($\neq 0$) $\in \mathbb{F}_q$ and computes Y = s + Xr. Computes shares of s and r using sss also and gives every player a pair (s_i, r_i).
Reconstruct: Each player sends (s_i', r_i') to the reconstructor R. R receives n shares and at most t are possibly incorrect. For each subset of t+1 players, R computes s' and r', checks if Y = s' + Xr'. If yes, R outputs the secret s'

References

[1] Paul Feldman. “A practical scheme for non-interactive verifiable secret sharing”. In: *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*. IEEE. 1987, pp. 427–438.

[2] Mahabir Prasad Jhanwar and Reihaneh Safavi-Naini. “On the share efficiency of robust secret sharing and secret sharing with cheating detection”. In: *International Conference on Cryptology in India*. Springer. 2013, pp. 179–196.

[3] Tal Rabin and Michael Ben-Or. “Verifiable secret sharing and multiparty protocols with honest majority”. In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. 1989, pp. 73–85.