A Multi Secret Sharing Scheme with Cheater Detection and Identification

Siddharth Bansal

Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India

Abstract

Secret Sharing, one of the fundamental aspects of key management, is a Cryptographic method that distributes a confidential piece of information among multiple entities. This adds an extra layer of security by necessitating collaboration for secret reconstruction. They can also be extended to share multiple secrets. As a result of the potential vulnerabilities in traditional secret-sharing schemes, Verifiable Secret Sharing(VSS) has been developed. VSS not only addresses issues of dishonesty within the sharing process but also introduces mechanisms for verifying the integrity of the shared secrets, enhancing the overall trustworthiness of secret-sharing schemes. Often these schemes do not maintain verification of all entities in the scheme or use private channels. Our proposed scheme for MSS enables the detection and identification of cheating by all entities and does not require private channels for the transfer of secrets as well making it lightweight.

Introduction and Prior Works

- In the world of information technology, the management of cryptographic keys is an important task, especially in the presence of multiple wardens none of whom are fully trustworthy.
- This motivates us to use Secret Sharing schemes which, instead of entrusting the entire key to one entity, distributes the key among different entities. This distributed responsibility provides enhanced security for the secret.
- In the realm of secret sharing algorithms, various methodologies have been proposed, each rooted in distinct mathematical concepts. Adi Shamir [1] introduced the Lagrange polynomial interpolation-based scheme, forming the well-known Shamir's Secret Sharing scheme. Other notable alternative approaches include the application of hyperplane geometry and the Chinese Remainder Theorem. These schemes collectively fall under the umbrella of (t, n) threshold secret sharing (TSS), where n represents the number of authorized participants (shareholders), and t signifies the minimum number of shares required to reconstruct the secret.
- In a Threshold Secret Sharing (TSS) scheme, the individual possessing the secret (the dealer) divides the secret S into n shares denoted as $\{s_1, s_2, ..., s_n\}$ and subsequently allocates these shares among authorized participants, each holding a single share. The critical feature of TSS lies in the requirement that the original secret S can only be reconstructed when at least t shares are collectively accessible and no information regarding the secret can be obtained from any set of less than t shares. The entity responsible for combining the shares, determining the threshold number of participants, and reconstructing the secret is known as the combiner. This combiner can either be one of the participants or a distinct trusted entity. This introduces adaptability to the secret-sharing procedure.

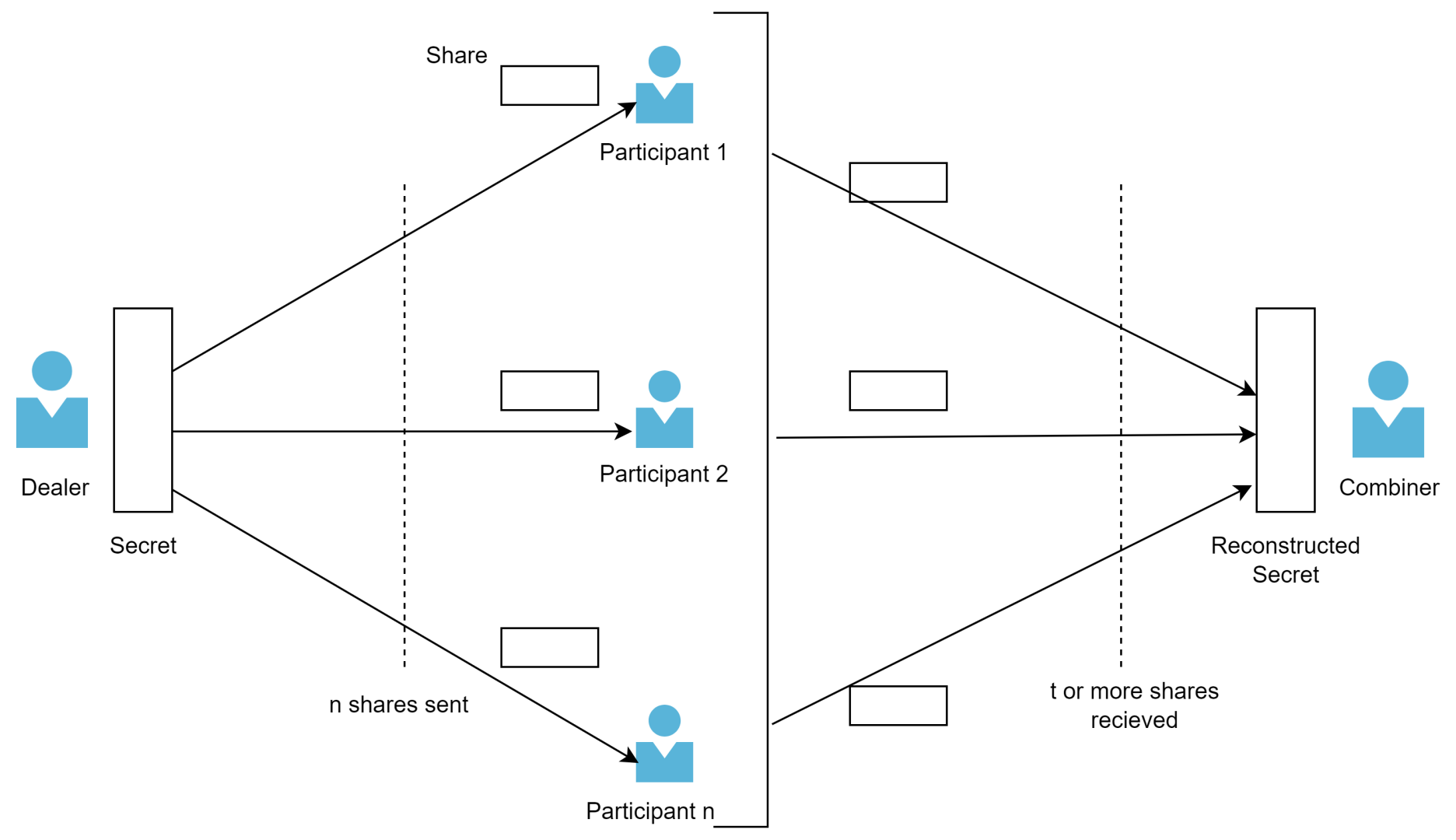

Figure: The Working of an (n, t) TSS Scheme

- TSS schemes can be extended to share multiple secrets at once. These Schemes are called Multi Secret Sharing(MSS) Schemes [2].
- In a real-world scenario, the dealer may act maliciously, participants may provide deceptive shares, or adversaries may pose as the combiner to gain unauthorized access. To deal with such acts, Verifiable Secret Sharing(VSS) Schemes were introduced.
- VSS schemes enable the participants to independently verify the accuracy of their shares and the overall reconstruction process, enhancing transparency and trust in the secret sharing protocol.
- Most VSS schemes allow for cheater Identification and Detection but only when the cheating is done by the dealer or the participants. The combiner is assumed to be honest.
- However, a malicious third party can impersonate the combiner and request shares from participants. If enough participants submit their shares to this fake combiner, the secret will be revealed.
- some schemes, like [3], allow for the identification of dishonest combiner but assume the dealer to be trusted and require the use of private channels which poses a practical limitation.
- All these factors prompted us to introduce a verifiable multi-secret sharing scheme designed to detect dishonest behavior across all three entities, all achieved without the reliance on private channels.

ECDLP and ECDHP

In our scheme, we are using the Elliptic Curve Discrete Logarithm problem(ECDLP) for the security of our scheme and the Elliptic Curve Diffie Helman protocol(ECDHP) for private communication over public channels. The ECDLP and ECDHP are described below

- The Elliptic Curve Discrete Logarithm problem(ECDLP) is as follows. Given an elliptic curve E and two points N_1 and N_2 on the curve, finding an integer n such that $nN_1 = N_2$ is called the Elliptic Curve Discrete Logarithm problem. ECDLP is computationally infeasible.
- The Elliptic Curve Diffie-Hellman (ECDH) protocol is an elliptic curve-based key exchange algorithm that enables two parties to independently generate a shared secret over an untrusted communication channel without explicitly exchanging the secret itself.
- The public parameters of ECDH are:
  - **n**: the order of the finite field Z_n.
  - **a and b**: constants that constitute the elliptic curve E.
  - **P**: a non-trivial point on the curve.
  - **H(.)**: a one-way collision-resistant hash function.
- The ECDH protocol operates as follows:
  - The two parties choose their secret keys (a for party A and b for party B) and compute their corresponding public keys (aP and bP) on the elliptic curve.
  - Party A publishes its private key aP and Party B publishes its private key bP.
  - They then use their private key and the public key of the other from the public channel to compute the mutual secret S = a(bP) = b(aP).
- The adversary has access to aP and bP(from the public channel). Using the security of the ECDLP, The values of a and b cannot be computed and thus it is unable to figure out the secret.

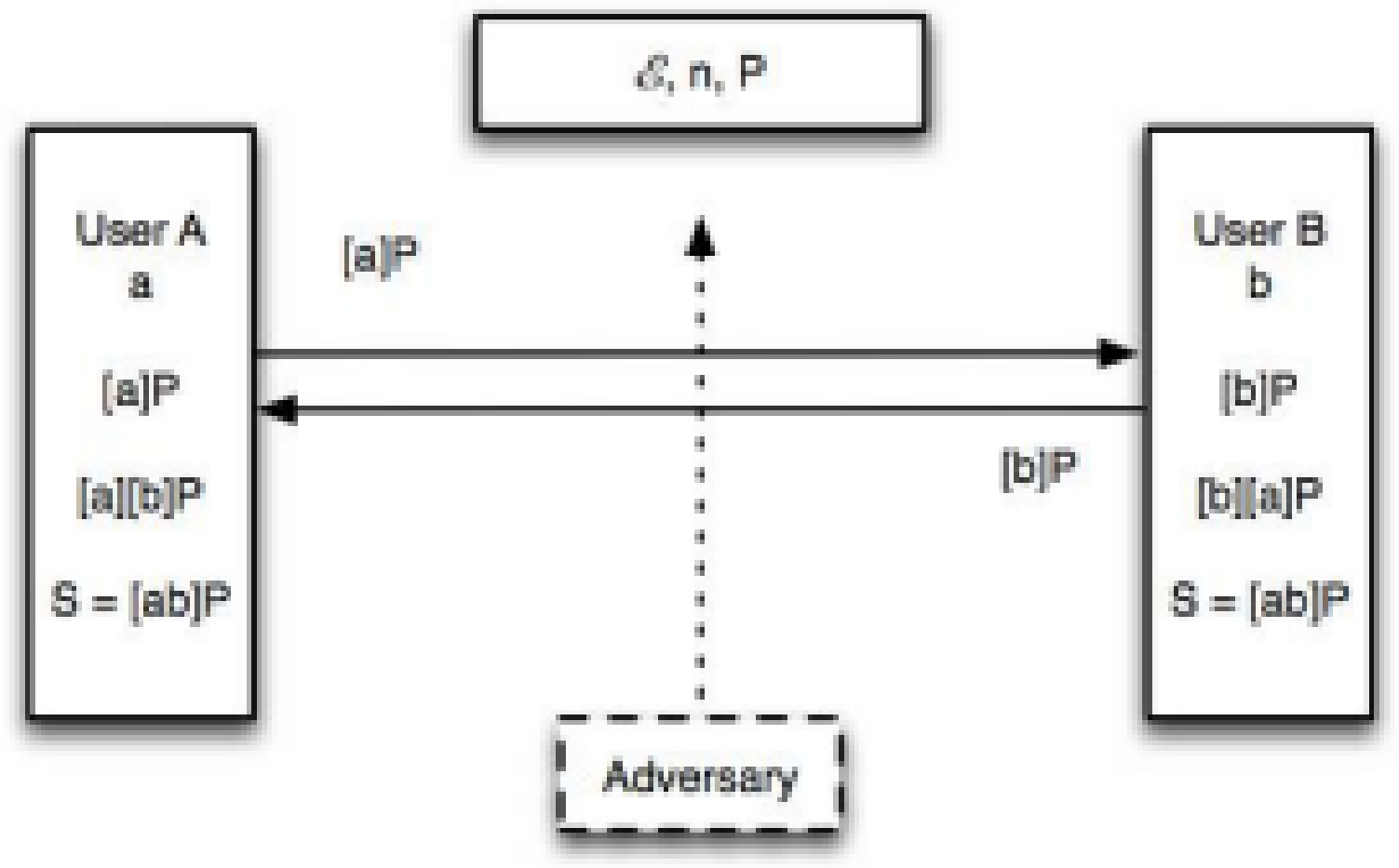

Figure: Elliptical Curve Diffie Helman Protocol

System Description

The domain parameters for our algorithm are:

- q: the order of the finite field Z_q
- two constants a and b which define the elliptic curve E
- G: a point on the curve
- m: the order of $< G >$ i.e. m is the least positive number satisfying $m.G = O$

We consider a secure Bulletin board (SBB) as a trusted entity that is used by the entities of our proposed scheme to publish keys and other public values in an authentic manner. We have assumed the following properties for the secured bulletin board:

- All the entities in the scheme are authenticated on the SBB.
- The entities' identities are verified every time they access the SBB.
- If a participant acts as the combiner as well, they will have to register with the SBB two separate times, once as the combiner and once as a participant.
- The SBB provides read/write access to all the entities after authentication. Although all the entities can read all the data published on the bulletin board, an entity can only modify or remove the entries it has uploaded.

Problem

The Dealer D has to share k secrets $\{S_1, S_2, ...S_k\}$ across n participants $P = \{P_1, P_2, ..., P_n\}$ such that any set of these shares with size >t can be used by the combiner C to decode the secrets.
Since none of the entities are fully trustable, we also have to implement methods for verification of the data sent by the entities.

Proposed Algorithm

- D publishes the domain parameters on the SBB.
- All the entities choose a private key and publish their public keys on the SBB. The SBB ensures that all keys are unique by asking the duplicate key owners to generate another.
- The dealer then selects a random number r for the sharing process.
- The dealer the establishes pseudo-shares $\{x_1, x_2, x_3, ..., x_n\}$ with participants and simultaneously establishes a secret with the combiner(ϕ_c) through the ECDH protocol. The pseudo-shares are a combination of the shared key formed during ECDH and the random r.
- Subsequently, the dealer selects pseudo secrets $s_1, s_2, ..., s_k$ corresponding to the secrets $\{S_1, S_2, ...S_k\}$ and conceals the secrets by adding them with both the pseudo secrets and the combiner's secrets. The results are then posted on the SBB.
- The encoding of pseudo secrets into n shares involves embedding them into a function and representing the shares as points on the function. Linear interpolation enables the reconstruction of pseudo secrets from a sufficient(>= t) number of shares.
- The combiner sends a submission request for pseudo shares to participants, accompanied by a verification message to ensure the request's authenticity.
- Participants generate secret keys for communication with the combiner using ECDH, using these keys to mask their shares. The combiner can subsequently decode these masked values using the secret keys.
- Pseudo secrets are reconstructed by the combiner from the received shares through linear interpolation. Leveraging these pseudo secrets and the combiner secret established earlier allows for the recovery of the original secrets.
- Since all the communication takes place over public channels, the threat of man-in-the-middle attacks is mitigated
- The verification for the keys generated using ECDH can be done as follows:
  - Let A and B be the parties and X be their shared key.
  - A chooses a verification number m. A then publishes the pair $(m.X, m)$.
  - B can then verify if the value shared by A is correct by calculating $m.X$ itself and checking if the values are the same.

Conclusion and Future Goals

- We introduced a threshold multi-secret sharing scheme with the ability to detect any cheating done by the dealer, participants, or combiner and also cheater identification.
- To ensure security, a secure bulletin board is used as a trusted authority, and all entities must be registered with it.
- No private channel was used between participants, dealers, and combiners to transfer any pseudo-secrets/secrets. This along with the use of ECC keeps the scheme lightweight in terms of resources.

Our plan is to implement the proposed algorithm in order to determine the actual time complexities of each step.
Additionally, we will attempt to verify the proposed verifiability of the entities by examining scenarios where cheating is perpetrated by the dealer, participants, or the combiner

References

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A (t,n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0096300303003552

[3] S. Kandar and B. C. Dhara, "A verifiable secret sharing scheme with combiner verification and cheater identification," *Journal of Information Security and Applications*, vol. 51, p. 102430, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212619304995

Acknowledgements

This work was done under the supervision of Prof. Sukumar Nandi and Mr. Saurav Gupta.

Contact Information

- Email: siddharth.bansal@iitg.ac.in