A Verifiable Threshold MSS Scheme Based on ECC

Sanchita Saha[a,b], Arup Kumar Chattopadhyay[c], Amitava Nag[a], Sukumar Nandi[d]

[a] *Central Institute of Technology Kokrajhar, India*
[b] *Haldia Institute of Technology, India*
[c] *Indian Institute of Technology Madras, India*
[d] *Indian Institute of Technology Guwahati, India*

Abstract

A multi-secret sharing (MSS) scheme enables the distribution of several secrets among a set of participants, such that only some well-defined subsets of participants are allowed to recover the secrets. A multi-secret sharing scheme is verifiable (VMSS) if *cheating detection* and *cheater identification* are conceivable in the secret sharing process. *Cheating detection* allows verification of the shares and/or secrets for their correctness, and notify the verifier in case of any mismatch. *Cheater identification* is a process of recognizing the entities that performed the cheating, and the cheater can be (1) the dealer who distributes incorrect share(s) to the participant(s), (2) the participant(s) who submit the fake share(s) to the combiner during the secret recovery process, or (3) the fake combiner (an intruder) who acts as the real combiner and tries to pool the shares from the participants with an intention to reveal the secret(s). In most of the VMSSs proposed, it is considered that the dealer and participants can be dishonest; however, the combiner is a fully trusted entity. Nevertheless, an intruder may attempt to impersonate the combiner and send a fake submission request to the participants, and by accumulating sufficient shares, he can compute the secrets. As per our knowledge, very limited studies are conducted on combiner verification. In this paper, we present a verifiable threshold multi-secret sharing

Email addresses: sanchita.cse2007@gmail.com (Sanchita Saha), ardent.arup@gmail.com (Arup Kumar Chattopadhyay), amitava.nag@cit.ac.in (Amitava Nag), sukumar@iitg.ac.in (Sukumar Nandi)

scheme that allows cheater identification and verification of the participants and the combiner. It also prevents the dealer from becoming a cheater. Additionally, it allows the use of public channels to exchange different information between the entities. The security aspect of the scheme which is proposed in this work is guaranteed under the *elliptic curve cryptography (ECC)* and the *elliptic curve discrete logarithm problem (ECDLP)*.

Keywords: secret sharing, threshold secret sharing, multi-secret sharing, verifiable secret sharing, elliptic curve cryptography

1. Introduction

In the domain of information security, secret sharing has evolved as a branch of research to secure sensitive information from misuse by unauthorized parties. Shamir [1] and Blakley [2] were the pioneer researchers for independently introducing two distinct secret sharing schemes. The premise of Shamir's [1] scheme is the Lagrange polynomial interpolation, whereas the Blakley [2] method is grounded on the idea of hyperplane geometry. Another secret sharing scheme, proposed by Mignotte [3] and which was improved by Asmuth and Bloom [4] is established on the Chinese remainder theorem (CRT). Afterward, many researchers [5, 6, 7], looking at a variety of different contexts, have extended the scheme proposed by Shamir [1]. Each of the ideas is a scheme for threshold secret sharing that is commonly defined as (t, n) threshold secret sharing (TSS). In a (t, n) TSS scheme, the owner of the secret or a reliable mediator, also known as the **dealer**, encodes a secret into n parts that are referred to as shares (or shadows). The shares are allocated among n number of authorized or approved **participants** in such a manner that each of the participants holds exactly one share. Reconstruction of the secret or secrets is achievable only when these approved participants submit their corresponding shares. In the (t, n) TSS scheme, t number (referred to as the threshold value) of shares are combined to regenerate the original secret. The right to choose a threshold number of participants, or more than that, and gather shares from those participants is granted to the

combiner. **Combiner** can be one who is selected from the participants of the scheme or a separate entity with the responsibility of reconstructing the secret from the gathered shares. However, the secret is not reconstructed if the count of shares is less than the threshold value t. There are several real-life applications of secret sharing schemes like secure data outsourcing in the cloud [8], in development of secure protocols and data transportation in IoT [9], blockchain applications [10], secure QR-code management [11], key management at ad-hoc networking, fair exchange [12], threshold proxy signature [13], e-voting [14], secure computing [15], e-Tendering [16], federated learning [17] and many more. Moreover, secret sharing can be used to share media files like images [18] and audio [19].

Multiple secrets can also be shared concurrently in a **multi-secret sharing scheme (MSS)**. He and Dawson proposed [20] a MSS scheme using a one-way function which reconstruct secrets in multiple stages. Chien et al. [21] proposed another MSS scheme which is based on symmetric block code and an one-way two-variable function. Some other popular MSS schemes are presented in [22, 23, 24, 25, 26, 27]. However, based on Shamir's TSS [1] scheme, Yang, Chang, and Hwang [28] presented a popular two-variable one-way function based MSS scheme, referred to as YCH scheme [28]. Compared to the Chien et al.'s [21] scheme, this scheme [28] uses less public values in the sharing process.

Shamir [1] assumes all three actors (dealer, participants, and combiner) in the secret sharing process are trustworthy. It means (i) dealers give participants the proper shares, (ii) share submission requests originate from a trusted combiner, and (iii) participants do not even submit false information. However, in real life scenarios, there is no guarantee that these criterion are always hold, as a result significant security risks happens during secret sharing process. There are several possibility of cheating as described by Zhao et al. [6]. It is possible that the dealer may send fake or incorrect shares to the participants (as shown in Figure 2), making it impossible for those participants to ever obtain the actual secret. If a participant acts as a cheater and sends fake share during the recovery process (as shown in Figure 3), real secret can not be reconstructed. An intruder

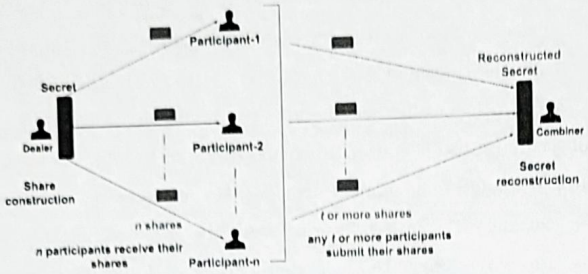

Figure 1: The process of (t, n)-TSS scheme.

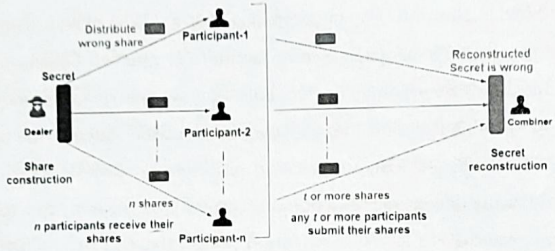

Figure 2: Cheating by dealer

may alter the shares during share distribution phase (as shown in Figure 4). An adversary may act as the combiner asking for the shares from participants (as shown in Figure 5), as a result actual secret is revealed at the wrong entity. Zhao et al. [6] broaden YCH scheme [28] into a verifiable multi-secret sharing (VMSS) scheme by using hardness of the discrete logarithm problem (DLP) which preserves both the share's and the secret's security. Subsequently, many verifiable secret sharing (VSS) schemes [29, 30, 31, 32, 33, 34, 35, 36, 37] were introduced for preventing Various forms of fraud in the secret sharing process.

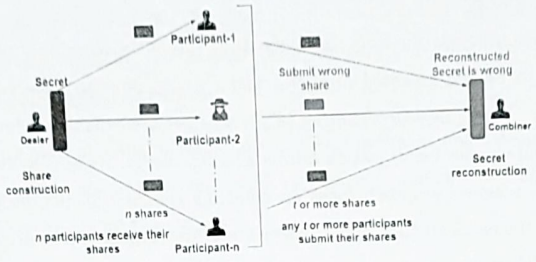

Figure 3: Cheating by participants.

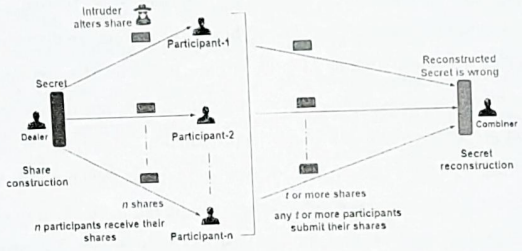

Figure 4: Intruder alters share

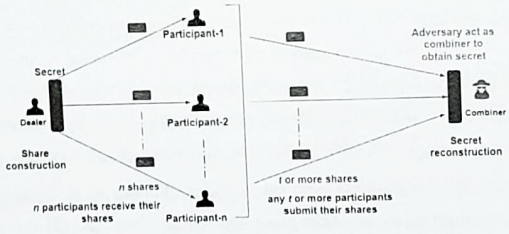

Figure 5: Cheating at combiner side

1.1. Motivation of the work

An obvious revelation from the above study is that the secret sharing schemes that are to be implemented in a practical scenario have to have verification properties. Verifiable secret sharing (VSS) schemes are capable of cheating detection and cheater identification during sharing secrets. As already discussed, most VSS schemes proposed focus on detecting cheating by the dealer or by one or more participants. Some schemes are also capable of identifying the cheater. The common assumption in these VSS schemes is that the combiner is either a participant or a trusted third party. However, a malicious entity outside the group of participants (or a fake combiner) may request the submission of the shares from the participants. If a predefined number of participants surrender their corresponding shares to the fake combiner, the secrets are disclosed. Multi-secret sharing schemes, such as [38, 39, 40] are based on the assumption that all entities involved in the secret sharing are honest and thus do not require the verification property. Although the works proposed in [41, 42, 43, 44, 45] implement cheating detection and cheater identification, these schemes lack combiner verification by the participants. The studies conducted in the field of combiner verification are very limited. Kandar and Dhara [37] presented a password-based authentication scheme for the combiner verification. However, some limitations of this scheme [37] are: (1) the dealer has to be a trusted entity; (2) the share distribution takes place over some secure channels, which is not practical; (3) it can share a single secret in each sharing process; and (4) the submission of the shares to the combiner also needs to take place over some secure channels. This motivates us to present a verifiable secret sharing scheme having cheater identification capability along with the following features: (1) it assumes none of the entities as fully trusted entities, as a result, verifies all the entities such as the dealer, the participants, and the combiner for any kind of cheating; (2) the distribution of shares can take place over public channels; (3) it is a multi-secret sharing scheme; and (4) the participants can submit their own shares to the combiner over public channels.

1.2. *Key contributions of the proposed scheme*

In this paper, a verifiable threshold multi-secret sharing scheme has been proposed based on *ECC* and *ECDLP*. The key contributions of the proposed scheme are highlighted as follows:

- We present a verifiable secret sharing scheme that enables verification of all the entities involved in the secret sharing process – the dealer, the participants, and the combiner.
  - It ensures that the dealer cannot become a cheater and send fake shares to the participants.
  - It ensures that if any participant involved in the process submits wrong or fake share to the combiner, the cheating can easily be detected and the cheater participant is identified.
  - It ensures that a request for sharing submissions to recover the actual secret coming from a combiner can be verified if the combiner is already registered (to a trusted center) in the given secret sharing system.
- The share distribution in the proposed schemes can be conducted over public channels.
- We propose a multi-secret sharing scheme that can share multiple secrets simultaneously in a single sharing process, and the reconstruction of all the secrets possible in a single stage.
- In the proposed work, the participants can submit their shares over public channels without disclosing any information about the secrets to any other participants or any malicious user outside the group.

The rest of the paper is arranged as follows: in Section 2, we study on some of the schemes that are closely relevant to the proposed scheme; in Section 3 we briefly discuss the basics of threshold secret sharing *(TSS)*, *ECC* and *ECDLP*; we present our proposed scheme in Section 4; the security analysis of

the presented scheme and comparison study with other works are presented in Section 5; and lastly, we conclude our work in Section 6.

2. Related Study

Traditional schemes for secret sharing make the assumption that that the dealer and all other participants are trustworthy (honest). In actuality, though, the dealer can be dishonest and give the participants some false shares. Participants with false shares are unable to access the real secrets. As opposed to that, dishonest participants can submit wrong or fake shares throughout the reconstruction process in order to mislead the other participants. The *verifiable secret sharing (VSS)* addresses these concerns regarding data security during the secret sharing process. Cheating detection and/or cheater identification is the primary objective of the VSS schemes. Chor et al. [46] first established a VSS scheme to detect cheating by the dealer throughout the secret sharing process. However, the scheme [46] is unable to identify malevolent participants since they are able to proceed with a fake share in order to produce the incorrect secret. This limitation is overcome by the scheme introduced by Harn and Lin [47] which has the ability to determine cheating and identify the dishonest participants. Stadler [48] proposed another VSS scheme that detects cheating by the dealer as well as by any participants involved in the scheme. Subsequently, Shao and Cao [49] proposed an efficient VMSS scheme on the basis of the YCH [28] scheme. However, in both the schemes [28, 49], the dealer selects the secret shadows. As a result, the dealer can become a cheater. A private channel is used for the information exchange between the dealer and the participants, which produce a high distribution cost.

Zhao et al. [6] presented a VMSS scheme (known as *ZZZ* scheme) based on *YCH* scheme [28], and Hwang and Chang's [50] scheme (*HC* scheme). The ZZZ scheme has the following advantages over the previous schemes: (i) It ensures that the cheater is identified, regardless of whether the cheater is a dealer or a participant; and (ii) Shares can be transmitted over a public channel rather than

a private channel since all the participants are able to select their own shadows for computing their corresponding shares. Dehkordi and Mashhadi [51] proposed another VMSS scheme based on the YCH scheme [28]. The authors added the verification property to it using the boldness of *discrete logarithm problem* and the *RSA cryptosystem*. Dehkordi and Mashhadi [52] presented two more VMSS schemes based on homogeneous linear recursion and public key cryptosystems such as RSA cryptosystem and Diffie Hellman scheme. Furthermore, Dehkordi and Mashhadi [30] proposed other two VMSS schemes based on non-homogeneous linear recursions and *elliptic curve cryptography (ECC)*. The security of the schemes is based on *elliptic curve RSA cryptosystem* and the boldness of the *ECDLP*. The VMSS schemes introduced in [52, 30] are computationally more efficient and secure than the ZZZ scheme [6] and the SC scheme [49]. Dehkordi and Oraei [53] presented a novel VMSS scheme with a general access structure and the security of the scheme depends on the graded encoding schemes. Das and Adhikari [54] introduced a different approach by using a one-way hash function so that both the participants and the combiner are capable to verify their own shares. Endurthi et al. [26] proposed a VMSS scheme based on Asmuth-Bloom's [4] scheme and Mignotte's [3] scheme. In this scheme, the shares acquired by the participants are reusable for a new set of secrets, and the shares issued by the dealer are verifiable by the participants. The verifiable secret sharing scheme proposed by Kandar and Dhara [37] along with the dealer and participants, can also verify the combiner. However, the scheme is not able to share multiple secrets simultaneously.

3. Preliminaries

The different abbreviations and symbols used in this paper are recorded in Table 1 and Table 2 respectively.

3.1. Threshold secret sharing (TSS)

The (t, n)-TSS is defined in its simplest form. Let the dealer $\mathcal{D}$ holds a secret S, and intends to share the secret among a group of n participants:

Abbreviations	Full Form
TSS	Threshold Secret Sharing
MSS	Multi-Secret Sharing
VMSS	Verifiable Multi Secret Sharing
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDH	Elliptic Curve Diffie Hellman
SBB	Secure Bulletin Board

Table 1: List of abbreviations

$\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$. The dealer $\mathcal{D}$ encodes the secret S into n parts called shares or shadows: $s_1, s_2, \cdots, s_n$, and distributes the shares in such a way that every participant gets precisely one share. Any t or more ($\leq n$) participants collaborate, and they can reconstruct the secret S. However, any group of less than t number of participants is unable to reveal any information about the secret. The formal definition of (t, n)-TSS scheme [55] presented is as follows:

Definition 3.1. A (t, n)-TSS scheme is a method of distributing a secret S among a set of participants $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$, in such a way that any t or more ($\leq n$) participants can recover the S, however, no group of $t-1$ or fewer participants are able to do.

Assume, the dealer $\mathcal{D}$ chooses the secret S, and $\mathcal{D} \notin \mathcal{P}$. The S secret is encoded and shared among the participants in such a way that none of the participants are aware to the shares that are obtained by the other participants. A subset of participants $p \subseteq \mathcal{P}$ pools their shares with an intention to compute the secret. Alternatively, the participants in p submit their shares to a trusted third-party called combiner $\mathcal{C}$ to commute the secret. The secret can be computed successfully if $|A| \geq t$, otherwise, the secret cannot be computed. Figure 1 depicts briefly the (t, n)-TSS process.

Symbol	Meaning
$\mathcal{D}$	Dealer
$\mathcal{C}$	Combiner
$\mathcal{P} = \{P_i\}_{i=1}^{n}$	Set of n participants
$\{S_i\}_{i=1}^{k}$	k number of secrets
$\{s_i\}_{i=1}^{k}$	k number of pseudo secrets
$H(\cdot)$	One-way hash function
$g(\cdot)$	Bi-variate one-way function
$\|\|$	Concatenation operator
$\mathbb{Z}_q$	A finite field with order q
$\mathcal{E}$	An elliptic curve $\in \mathbb{Z}_q$
G	$G \in \mathbb{Z}_q$, generator of the cyclic group $\langle G \rangle$
m	Order of $\langle G \rangle$ satisfying $m \cdot G = \mathcal{O}$
$a_i,\ 1 \geq i \geq n$	Private key (or *secret shadow*) of i^{th} participant
a_c	Private key of combiner
a_0	Private key of dealer
$A_i,\ 1 \geq i \geq n$	Public key of i^{th} participant
A_c	Public key of combiner
A_0	Public key of dealer
$PID_i,\ 1 \geq i \geq n$	Identification number of i^{th} participant
DID	Identification number of dealer
CID	Identification number of the combiner
Φ_c	Secret key shared between dealer and combiner
$X_i,\ 1 \geq i \geq n$	Pseudo share of i^{th} participant computed by $\mathcal{D}$
$Y_i,\ 1 \geq i \geq n$	Public share of i^{th} participant
$v_i,\ 1 \geq i \geq n$	Public verifier for each $P_i \in \mathcal{P}$
$v_i^{(c)}$	Combiner verifier
Z_i	Public information published by $\mathcal{D}$
T_i	Timestamp

Table 2: List of symbols

3.2. *Primary Entities of a Threshold Secret Sharing (TSS) scheme*

A TSS scheme is made up of the following primary entities:

- *Secret / secrets:* A secret S or a set of secrets $\{S_1, S_2, \cdots, S_k\}$ is/are the various types of data, such as integers, texts, images distributed among a set of participants.

- *Shares and Shadows:* The secret/secrets are encoded into n shares, so that none of the individual shares disclose any information regarding the secrets.

- *Dealer:* Dealer $\mathcal{D}$ is responsible for encoding the secret/secrets into n shares/shadows and allocating those shares to the participants such that each participant receives exactly one share/shadow. The dealer is almost always the legitimate proprietor of the secret or secrets, or a mediator who can be trusted.

- *Participants:* Set $\mathcal{P} = \{P_i\}_{i=1}^{n}$ is used to represent the participants, who obtain the secret/secrets from the dealer.

- *Combiner:* A combiner $\mathcal{C}$ is in charge for decoding the secret/secrets if an approved subset of participants submits their own shares/shadows.

3.3. *A brief review of Elliptic Curve Cryptography (ECC)*

The *elliptic curve cryptography (ECC)* [55] and *elliptic curve discrete logarithm problem (ECDLP)* ensures the security of the proposed scheme. This section provides a basic introduction to these topics. Antipa et al. [56] covered some of standardized elliptic curve key generation methods and public-key encryption protocols, as well as various attacks that could be used against the protocols.

3.3.1. *Elliptic curves modulo a prime*

ECC uses elliptic curves in which the variables and coefficients are restricted to the elements of a finite field. Elliptic curves over $\mathbb{Z}_q$ can be defined as follows:

Let $q > 3$ be a prime. The *elliptic curve* $\mathcal{E}$ over the prime field $\mathbb{Z}_q$, presented as $\mathcal{E}(\mathbb{Z}_q)$, is the set of points $(x, y) \in \mathbb{Z}_q \times \mathbb{Z}_q$ satisfying the equation

$$y^2 = x^3 + \alpha x + \beta \ (mod\ q), \tag{1}$$

such that $4\alpha^3 + 27\beta^2 \neq 0$ (Let $\alpha, \beta \in \mathbb{Z}_q$ be two constant). This guarantees that the curve is non-singular and has distinct roots. If a specific point $\mathcal{O}$ called *point at infinity* is added, then it forms an additive algebraic group. Hence, $\mathcal{E}(\mathbb{Z}_q)$ is given by

$$\mathcal{E}(\mathbb{Z}_q) = \{(x, y) | x, y \in \mathbb{Z}_q \ and\ y^2 = x^3 + \alpha x + \beta \ (mod\ q)\} \cup \{\mathcal{O}\}. \tag{2}$$

The points of the elliptic curve $\mathcal{E}(\mathbb{Z}_q)$ form an abelian group as follows:

- $\mathcal{O}$ is the additive identity element. As a result, $\forall N \in \mathcal{E}(\mathbb{Z}_q)$, $N + \mathcal{O} = N = \mathcal{O} + N$.
- Let $N_1 = (x_1, y_1)$ and $N_2 = (x_2, y_2)$ be two points on $\mathcal{E}(\mathbb{Z}_q)$, such that $N_1 \neq -N_2$. If $x_1 = x_2$ and $y_2 = -y_1$, $N_1 + N_2 = \mathcal{O}$; else $N_1 + N_2 = (x_3, y_3)$, where:

$$x_3 = \lambda^2 - x_1 - x_2, \tag{3}$$

$$y_3 = \lambda(x_1 - x_3) - y_1, \tag{4}$$

and

$$\lambda = \begin{cases} \dfrac{3x_1^2 + a}{2y_1}, & \text{if } N_1 = N_2 \\ \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{if } N_1 \neq N_2 \end{cases} \tag{5}$$

3.3.2. Elliptic curve discrete logarithm problem (ECDLP) in ECC

Let N_1 and N_2 represent any two points on an elliptic curve $\mathcal{E}$ with prime order m, such that $N_2 = tN_1$. Given N_1 and N_2, anyone has to recover the multiplier t. The problem of finding out the t is defined as the *elliptic curve discrete logarithm problem*. In other words, computationally, it is not possible to find out the $t \in \mathbb{Z}_q^*$ [57].

3.3.3. Public key cryptography using elliptic curve

In *asymmetric cryptography/public key cryptography*, the communicating parties share some common parameters known as *domain parameters*. Let the domain parameters defined by $(q, \alpha, \beta, G, m, h_c)$, comprise the following:

- q denotes the order of the finite field $\mathbb{Z}_q$,
- There are two constants $\alpha, \beta \in \mathbb{Z}_q$ that define the elliptic curve $\mathcal{E}$, so that $4\alpha^3 + 27\beta^2 \neq 0$,
- $G = (x_G, y_G) \in \mathbb{Z}_q$ is the generator of the cyclic group $\langle G \rangle$,
- m is the order of $\langle G \rangle$ so that m is the least positive number satisfying $m \cdot G = \mathcal{O}$,
- $h_c = \frac{1}{m}|\mathcal{E}(\mathbb{Z}_q)|$ is the cofactor.

3.3.4. Elliptic curve Diffie-Hellman protocol (ECDHP) in ECC

Elliptic-curve Diffie Hellman is a anonymous key agreement scheme that permits two involving parties, such as P_1 and P_2 having elliptic-curve public private key pairs to agree on a shared secret across an insecure channel. ECDHP is very similar to the traditional Diffie Hellman Key Exchange algorithm, with the exception that it employs ECC point multiplication rather than modular exponentiation. $H(\cdot)$ is assumed as an one-way collision-resistant hash function. The following steps illustrate how a shared key is established:

1. P_1 chooses an integer $a_{P_1} \in \mathbb{Z}_q^*$ as its private key, then calculates its public key A_{P_1} as $A_{P_1} = a_{P_1}G$. The private-public key-pair of P_1 is denoted as (a_{P_1}, A_{P_1}).
2. P_2 chooses an integer $a_{P_2} \in \mathbb{Z}_q^*$ as its private key, then calculates its public key A_{P_2} as $A_{P_2} = a_{P_2}G$. The private-public key-pair of P_2 is denoted as (a_{P_2}, A_{P_2}).
3. P_1 sends A_{P_1} to P_2 and P_2 sends A_{P_2} to P_1. Assume P_1 and P_2 gets the original version of each other's public key.

4. P_2 computes the secret key $K_1 = a_{P_2} A_{P_1}$ and $k_1 = H(x(K_1))$. P_1 computes the secret key $K_2 = a_{P_1} A_{P_2}$ and $k_2 = H(x(K_2))$

The shared secret key computed by P_1 and P_2 are equal, because:

$$\begin{aligned} k_1 &= H(x(a_{P_2} A_{P_1}) = H(x(a_{P_2}(a_{P_1} G))) \\ &= H(x(a_{P_1}(a_{P_2} G))) = H(x(a_{P_1} A_{P_2})) = k_2. \end{aligned}$$

P_1 initially exposes only its public key A_{P_1}. So, no one except P_1 can determine the private key a_{P_1}, unless that party can solve the *ECDLP*. Similarly, the private key a_{P_2} of P_2 is secure. No one other than P_1 or P_2 can compute the shared key, unless that party can solve the *ECDHP*.

3.3.5. Validation of public key

The public key $A = (x_A, y_A)$ of an elliptic curve is considered valid if it holds the following: (i) $A \neq \mathcal{O}$, (ii) x_A and y_A are properly represented elements in $\mathbb{Z}_q$, (iii) A satisfies the define elliptic curve $\mathcal{E}$ and (iv) $mA = \mathcal{O}$

3.3.6. The reason for choosing elliptic curve cryptography (ECC) instead of conventional encryption techniques :

Elliptic curve cryptography (ECC) depends on groups made out of elliptic curve points. The groups of elliptic curves are more desirable from a cryptography standpoint for the following reasons:

- Discrete Logarithm problem and Diffie-Hellman cryptosystems executed using elliptic curve groups are far secure than the executions based on prime-order subgroups of $\mathbb{Z}_q^*$. This is because there is no familiar sub-exponential time algorithm for solving the discrete logarithm problem in elliptic curve groups when the domain parameters are specified suitably.
- *ECC*-based encryption schemes use keys that are significantly smaller in size than non-*ECC* encryption algorithms (such as RSA), however, offer the same or greater security.
- Applications of *ECC* can be found in devices that have limited memory and power.

4. Proposed Scheme

In this section, we propose a verifiable threshold multi-secret sharing scheme. The proposed method includes six phases: 1) Initialization, 2) Verification of pseudo shares, 3) Share generation and distribution, 4) Combiner verification, 5) Transfer of the pseudo shares to the combiner, and 6) Secret reconstruction. Figure 6 depicts the block diagram of the proposed scheme. The different phases are described as below:

4.1. Initialization

Let $\mathcal{D}$ represents the dealer, $\mathcal{C}$ represents the combiner, and $\mathcal{P} = \{P_i\}_{i=1}^{n}$ represents the set of n participants.

- $\mathcal{D}$ chooses an elliptic curve $\mathcal{E}$ over $\mathbb{Z}_q$ in such a way that the domain parameters are $(q, \alpha, \beta, G, m, h_c)$, where:
  - q denotes the order of the finite field $\mathbb{Z}_q$,
  - There are two constants $\alpha, \beta \in \mathbb{Z}_q$ that define the elliptic curve $\mathcal{E}$, so that $4\alpha^3 + 27\beta^2 \neq 0$,
  - $G = (x_G, y_G) \in \mathbb{Z}_q$ is the generator of the cyclic group $\langle G \rangle$,
  - m is the order of $\langle G \rangle$ so that m is the least positive number satisfying $m \cdot G = \mathcal{O}$,
  - $h_c = \frac{1}{m}|\mathcal{E}(\mathbb{Z}_q)|$ is the cofactor.

We consider a *secure bulletin board* (SBB) as a trusted entity that is used by the entities of our proposed secret sharing scheme to publish the public keys and other public values required in the scheme in an authenticated manner. The following are the assumptions that we have considered for the *secure bulletin board* (SBB).

- All the entities that are participating in the secret sharing process, i.e., the dealer, the participants, and the combiner, all need to be registered

on SBB and are provided with appropriate credentials (user ID and password). The entities' identities are verified (entity authentication process) every time they access the SBB. We also assume the following to identify each entity:

- Let $DID \in \mathbb{Z}_m$ be the integer that uniquely identify the dealer.
- Let $PID_1, PID_2, \cdots, PID_n \in \mathbb{Z}_m$ be the integers that uniquely identify the participants, $P_1, P_2, \cdots, P_n$, respectively.
- Let $CID \in \mathbb{Z}_m$ be the integer that uniquely identifies the combiner. We consider exactly one combiner for each sharing process. The combiner can be chosen as one of the participants or a third-party duly registered to SBB for computing the secrets. If a participant wants to become a combiner, it has to separately register itself with the SBB as a combiner, and it assumes both the roles of a participant (with PID_i) and a combiner (with CID).

- SBB provides read-write access to all the entities after appropriate authentication. Although all the entities can read all the data published on the bulletin board, an entity can modify or remove only the entries that it has uploaded.

The entities securely exchange the pseudo shares using the ECDH protocol as follows:

1. First, $\mathcal{D}$ publishes the elliptic curve domain parameters $(q, \alpha, \beta, G, m, h_c)$ on SBB.

2. $\mathcal{D}$ chooses a cryptographic hash function, $H : \{0,1\}^* \to \mathbb{Z}_m$ that is secure and a bi-variate secure one-way function $g : \{0,1\}^* \times \mathbb{Z}_q \to \mathbb{Z}_q$. $\mathcal{D}$ makes both $H(\cdot)$ and $g(\cdot, \cdot)$ public.

3. Each participant $P_i \in \mathcal{P}$ chooses an integer $a_i \in \mathbb{Z}_m$ as its private key (or *secret shadow*), then computes the public key $A_i = a_i \cdot G$. Each P_i sends a request to SBB for publishing $\{PID_i, A_i\}$. SBB verifies that $A_i \neq A_j$

for all $j \neq i$; otherwise demands another public key from that participant. (to ensures that secret shadows are unique). Once the A_i is unique, it publishes the $\{PID_i, A_i\}$ on SBB.

4. $\mathcal{D}$ chooses an integer $a_0 \in \mathbb{Z}_m$ as its private key and computes the public key $A_0 = a_0 \cdot G$. $\mathcal{D}$ publishes $\{DID, A_0\}$ on SBB.

5. $\mathcal{C}$ also chooses an integer $a_c \in \mathbb{Z}_m$ as its private key and computes the public key $A_c = a_c \cdot G$. $\mathcal{C}$ publishes $\{CID, A_c\}$ on SBB.

6. $\mathcal{D}$ chooses a new random integer r for every sharing process and also publishes it on SBB.

7. $\mathcal{D}$ and $\mathcal{C}$ individually compute a secret key, Φ_c (which would be called as *combiner's secret*) as follows:

   7.1. $\mathcal{D}$ computes as follows:

      7.1.1. Compute $\phi_c = a_0 \cdot A_c$ as a point on $\mathcal{E}$.

      7.1.2. Compute $\Phi_c = g(r, \phi_c)$ as a point on $\mathcal{E}$.

   7.2. $\mathcal{C}$ computes as follows:

      7.2.1. Compute $\phi_c = a_c \cdot A_0$ as a point on $\mathcal{E}$.

      7.2.2. Compute $\Phi_c = g(r, \phi_c)$ as a point on $\mathcal{E}$.

8. $\mathcal{D}$ computes the pseudo shares for each participant $P_i \in \mathcal{P}$ as follows:

   8.1. Compute $B_i = a_0 \cdot A_i$, where B_i is a point on $\mathcal{E}$.

   8.2. Compute $I_i = g(r, B_i)$, where I_i is a point (x_{I_i}, y_{I_i}) on $\mathcal{E}$.

   8.3. Compute $X_i = H(x_{I_i} \oplus y_{I_i})$, where X_i is the pseudo share of P_i.

9. P_i computes its pseudo share X_i' as follows:

   9.1. Compute $B_i' = a_i \cdot A_0$, where B_i is a point on $\mathcal{E}$.

   9.2. Compute $I_i' = g(r, B_i')$, where I_i' is a point $(x'_{I_i'}, y'_{I_i'})$ on $\mathcal{E}$.

   9.3. Compute $X_i' = H(x'_{I_i'} \oplus y'_{I_i'})$.

4.2. Verification of pseudo shares

Each $P_i \in \mathcal{P}$ can verify the correctness of the computed pseudo share ($X_i \stackrel{?}{=} X_i'$) as follows:

1. $\mathcal{D}$ chooses another random number $\gamma_i \in \mathbb{Z}_m^*$ for each $P_i \in \mathcal{P}$ and compute $\Gamma_i = \gamma_i \cdot G$.
2. $\mathcal{D}$ publishes the public verifier v_i for each $P_i \in \mathcal{P}$ as follows:
   2.1. Compute $u_i = \gamma_i + h_i \cdot a_0$ where $h_i = H(X_i||PID_i||\Gamma_i)$.
   2.2. Publish $v_i = \{u_i, \Gamma_i\}$.

Any participant or combiner can verify the computed pseudo share's validity (i.e. $X_i' = X_i$) as follows:

$$u_i \cdot G \stackrel{?}{=} \Gamma_i + h_i' \cdot A_0, \text{, where } h_i = H(X_i'||PID_i||\Gamma_i)$$

Proof.

$$\begin{aligned} u_i \cdot G &= (\gamma_i + h_i \cdot a_0) \cdot G \\ &= \gamma_i \cdot G + h_i \cdot a_0 \cdot G \\ &= \Gamma_i + h_i \cdot A_0 \end{aligned}$$

Now, the equality $\Gamma_i + h_i \cdot A_0 = \Gamma_i + h_i' \cdot A_0$ ensures that $hi = h_i'$. Since $h_i = H(X_i||PID_i||\Gamma_i)$ and $h_i' = H(X_i'||PID_i||\Gamma_i)$, $X_i = X_i'$. □

A similar technique can also be formulated for the verification of the correctness of Φ_c by $\mathcal{C}$.

4.3. Share generation and distribution

$\mathcal{D}$ performs the following tasks.

1. Choose k secrets $\{S_i\}_{i=1}^k \in \mathbb{Z}_\eta$ which are points on $\mathcal{E}$.
2. Choose k pseudo secrets $\{s_i\}_{i=1}^k \in \mathbb{Z}_m$.
3. Compute $W_1 = s_1 \cdot G$, $W_2 = s_2 \cdot G$, $\cdots$, $W_k = s_k \cdot G$.

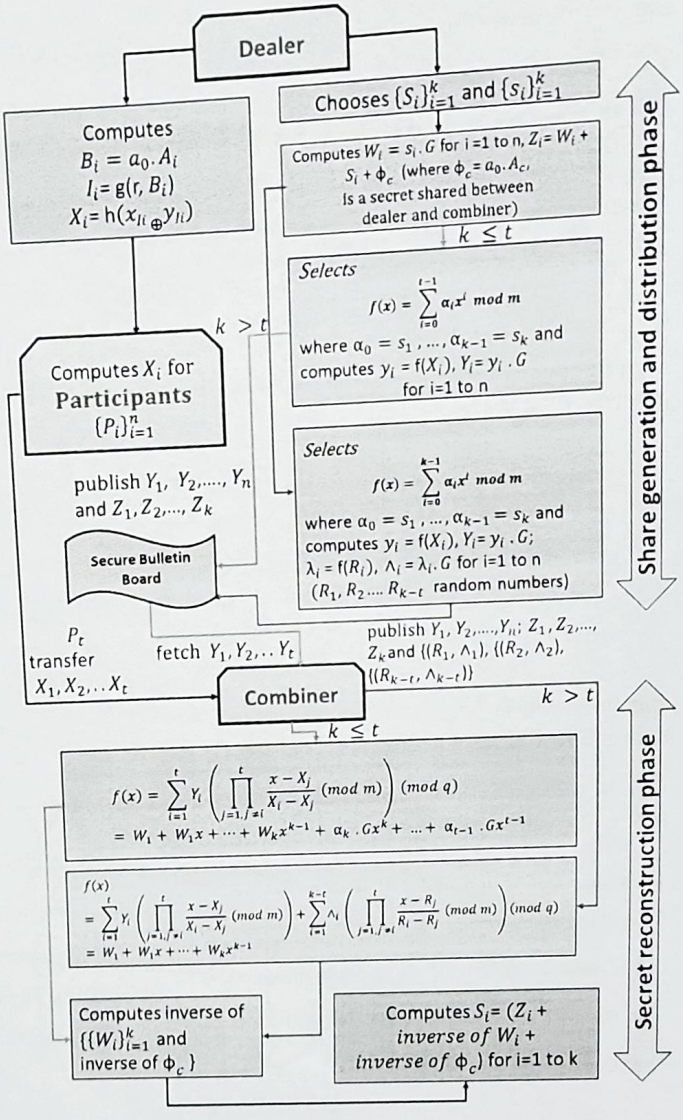

Figure 6: Block diagram of the proposed scheme

4. Compute $Z_1 = (W_1 + S_1 + \Phi_c), Z_2 = (W_2 + S_2 + \Phi_c), \cdots, Z_k = (W_k + S_k + \Phi_c)$ (all the operations are in $\mathbb{Z}_q$), and publish $\{Z_1, Z_2, \cdots, Z_k\}$ on SBB in authenticated manner.

5. Select a polynomial $f(x)$ as follows:

5.1. ***If*** $k \leq t$: $f(x)$ is a polynomial of $(t-1)^{\text{th}}$ degree, as shown below.

$$\begin{aligned} f(x) &= \sum_{i=0}^{t-1} \alpha_i x^i \ (mod\ m) \\ &= \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_{t-1} x^{t-1} \ (mod\ m) \end{aligned} \tag{6}$$

where $\alpha_0 = s_1, \alpha_1 = s_2, \cdots, \alpha_{k-1} = s_k$ and $\alpha_k, \cdots, \alpha_{t-1} \in \mathbb{Z}_m$ are the random integers selected by $\mathcal{D}$.

5.1.1. Compute the public values as follows:

(i) $y_i = f(X_i)$ for $i = 1$ to n,

(ii) $Y_i = y_i \cdot G$ for $i = 1$ to n.

5.1.2. Publish $\{Y_1, Y_2, \cdots, Y_n\}$ and $\{Z_1, Z_2, \cdots, Z_k\}$.

5.2. ***If*** $k > t$: $f(x)$ is a polynomial of $(k-1)^{\text{th}}$ degree, as shown below.

$$\begin{aligned} f(x) &= \sum_{i=0}^{k-1} \alpha_i x^i \ (mod\ m) \\ &= \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_{k-1} x^{k-1} \ (mod\ m) \end{aligned} \tag{7}$$

where $\alpha_0 = s_1, \alpha_1 = s_2, \cdots, \alpha_{k-1} = s_k$.

5.2.1. Choose $k - t$ random integers $R_1, R_2, \cdots, R_{k-t} \in \mathbb{Z}_m$ such that $\{X_1, X_2, \cdots, X_n\} \cap \{R_1, R_2, \cdots, R_{k-t}\} = \emptyset$

5.2.2. Compute the public values as follows:

(i) $y_i = f(X_i)$ for $i = 1$ to n,

(ii) $\lambda_i = f(R_i)$ for $i = 1$ to $k - t$,

(iii) $Y_i = y_i \cdot G$ for $i = 1$ to n,

(iv) $\Lambda_i = \lambda_i \cdot G$ for $i = 1$ to n.

5.2.3. Publish $\{Y_1, Y_2, \cdots, Y_n\}, \{(R_1, \Lambda_1), (R_2, \Lambda_2), \cdots, (R_{k-t}, \Lambda_{k-t}))\}$ and $\{Z_1, Z_2, \cdots, Z_k\}$.

4.4. *Combiner verification*

The combiner, $\mathcal{C}$ sends the submission request (for pseudo shares) to each participant $P_i \in \mathcal{P}$ along with a verification message so that P_i can verify if the request is coming for an authorized combiner only.

1. $\mathcal{C}$ creates a message $\{v_i^{(c)}, T_i\}$ to verify its identity with P_i (for $i = 1$ to n), where T_i is the timestamp and $v_i^{(c)} = a_c \cdot A_i \cdot H(PID_i||CID||T_i)$.
2. $\mathcal{C}$ transmits $\{v_i^{(c)}, T_i\}$ to each $P_i \in \mathcal{P}$.
3. P_i verifies it by validating the equality $v_i^{(c)} \stackrel{?}{=} (A_c \cdot a_i \cdot H(PID_i||CID||T_i)$. The equality proves the identity of the combiner.

Proof.

$$\begin{aligned} v_i^{(c)} &= a_c \cdot A_i \cdot H(PID_i||CID||T) \\ &= a_c \cdot a_i \cdot G \cdot H(PID_i||CID||T) \\ &= a_c \cdot G \cdot a_i \cdot H(PID_i||CID||T) \\ &= A_c \cdot a_i \cdot H(PID_i||CID||T) \end{aligned}$$

□

4.5. *Transfer of the pseudo shares to the combiner*

Our proposed scheme does not use any secret channel for transferring the pseudo shares to the combiner. Without loss of generality, it is assumed that $\{P_i\}_{i=1}^{t}$ are interested to compute the secrets, and they can do so by communicating over some public channel.

1. Each $P_i \in \mathcal{P}$ interested in computing the secrets performs the following tasks.
   1.1. Compute $D_i = a_i \cdot A_c$,
   1.2. Compute $K_i = I_i - D_i = I_i + D_i^{-1}$ and publish K_i on the SBB in an authenticated manner.

2. $\mathcal{C}$ computes pseudo shares X_i' by performing the following tasks.

   2.1. Compute $D_i' = a_c \cdot A_i$,

   2.2. Compute $I_i' = D_i' + K_i$ where K_i is available on SBB,

   2.3. Compute $X_i' = H(x_{I_i'} \oplus y_{I_i'})$.

3. $\mathcal{C}$ is able to verify the correctness of $X_i' = X_i$ using public verifier v_i by testing the following equality.

$$u_i \cdot G \stackrel{?}{=} \Gamma_i + h_i' \cdot A_0 \text{ where } h_i = H(X_i' || PID_i || \Gamma_i).$$

4.6. *Secret reconstruction phase*

After verifying the validity of t valid pseudo shares $\mathcal{C}$ performs the following tasks.

1. For each of the pseudo shares $X_1, X_2, \cdots, X_t$ fetch the corresponding public shares $Y_1, Y_2, \cdots, Y_t$ and other public values from SBB.

2. Applying Lagrange interpolation, the polynomial $f(x)$ can be reconstructed with the following conditions:

   If $k \leq t$:

$$\begin{aligned}
f(x) &= \sum_{i=1}^{t} Y_i \left(\prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \ (mod\ m) \right) (mod\ q) \\
&= \sum_{i=1}^{t} y_i \cdot G \left(\prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \ (mod\ m) \right) (mod\ q) \\
&= \sum_{i=1}^{t} y_i \left(\prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \ (mod\ m) \right) \cdot G \ (mod\ q) \\
&= \alpha_0 + \alpha_1 x + \cdots + \alpha_{k-1} x^{k-1} + \alpha_k x^k + \cdots + \alpha_{t-1} x^{t-1}) \cdot G \ (mod\ q) \\
&= (s_1 + s_2 x + \cdots + s_k x^{k-1} + \alpha_k x^k + \cdots + \alpha_{t-1} x^{t-1} \ mod\ m) \cdot G \ (mod\ q) \\
&= s_1 \cdot G + s_2 \cdot Gx + \cdots + s_k \cdot Gx^{k-1} + \alpha_k \cdot Gx^k + \cdots + \alpha_{t-1} \cdot Gx^{t-1} \ \ (mod\ q) \\
&= W_1 + W_2 x + \cdots + W_k x^{k-1} + \alpha_k \cdot Gx^k + \cdots + \alpha_{t-1} \cdot Gx^{t-1}
\end{aligned}$$

If $k > t$:

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{t} Y_i \left(\prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \ (mod\ m) \right) + \sum_{i=1}^{k-t} \Lambda_i \left(\prod_{j=1, j\neq i}^{t} \frac{x - R_j}{R_i - R_j} \ (mod\ m) \right) \ (mod\ q) \\
&= \sum_{i=1}^{t} y_i \cdot G \left(\prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \ (mod\ m) \right) + \sum_{i=1}^{k-t} \lambda_i \cdot G \left(\prod_{j=1, j\neq i}^{t} \frac{x - R_j}{R_i - R_j} \right) \ (mod\ q) \\
&= \sum_{i=1}^{t} y_i \left(\prod_{j=1, j\neq i}^{t} \frac{x - X_j}{X_i - X_j} \ (mod\ m) \right) + \sum_{i=1}^{k-t} \lambda_i \left(\prod_{j=1, j\neq i}^{t} \frac{x - R_j}{R_i - R_j} \ (mod\ m) \right) \cdot G \ (mod\ q) \\
&= (\alpha_0 + \alpha_1 x + \cdots + \alpha_{k-1} x^{k-1}) \cdot G \ (mod\ q) \\
&= (s_1 + s_2 x \cdots + s_k x^{t-1}) \cdot G \ (mod\ q) \\
&= s_1 \cdot G + s_2 \cdot Gx \cdots + s_k \cdot Gx^{t-1} \ (mod\ q) \\
&= W_1 + W_2 x + \cdots + W_k x^{k-1}
\end{aligned}
$$

3. Compute $W_1^{-1}, W_2^{-1}, \cdots, W_k^{-1} \in \mathbb{Z}_q$ from $W_1, W_2, \cdots, W_n$ available on SBB, and also compute Φ_c^{-1}.

4. Compute the secrets $S_i = Z_i + W_i^{-1} + \Phi_c^{-1}$ for $i = 1$ to k (all the operations are in $\mathbb{Z}_q$).

5. Security Analysis

Cheating detection and cheater identification are crucial security requirements for any secret sharing scheme, which are achieved in our proposed scheme using *ECC*. In this section, we scrutinize our proposed scheme against a lot of security attacks to demonstrate that it can withstand all of them.

- *Attack 1: (Dealer cheating) The dealer may transmit incorrect shares (or pseudo shares) in order to cheat one or more participants.*

  Analysis: In the proposed scheme, the dealer and each participant compute their pseudo share individually. As a result, the dealer cannot become a cheater. However, any participant is able to verify the validity of computed pseudo share (i.e. $X_i' = X_i$) as follows:

$$u_i \cdot G \stackrel{?}{=} \Gamma_i + h_i' \cdot A_0, \text{, where } h_i = H(X_i' || PID_i || \Gamma_i)$$

Proof.

$$\begin{aligned} u_i \cdot G &= (\gamma_i + h_i \cdot a_0) \cdot G \\ &= \gamma_i \cdot G + h_i \cdot a_0 \cdot G \\ &= \Gamma_i + h_i \cdot A_0 \end{aligned}$$

Now, the equality $\Gamma_i + h_i \cdot A_0 = \Gamma_i + h_i' \cdot A_0$ ensures that $hi = h_i'$. Since $h_i = H(X_i||PID_i||\Gamma_i)$ and $h_i' = H(X_i'||PID_i||\Gamma_i)$, $X_i = X_i'$. □

Therefore, the dealer $\mathcal{D}$ cannot become a cheater in the proposed scheme.

- *Attack 2: (Participant cheating) One or more participants may deposit fake shares to the combiner to cheat other participants.*

  Analysis: Suppose, one or more participants $P_i \in \mathcal{P}$ submit incorrect share X_i' ($X_i' \neq X_i$) to the combiner to cheat other participants. The X_i is computed by $\mathcal{D}$, and $\mathcal{D}$ also has published the verifier $v_i = \{u_i, \Gamma_i\}$ (where $u_i = \gamma_i, h_i.a_0$ and $h_i = H(X_i||PID_i||\Gamma_i)$) for P_i so that anyone has access to SBB can verify the pseudo share X_i' submitted by participant P_i. As a result, the verification of X_i' submitted by P_i fails as ($X_i' \neq X_i$).

- *Attack 3: (Participant attack) An intruder may impersonate a legal participant and gain some knowledge about the secrets.*

  Analysis: Any adversary can impersonate as a legitimate participant in the secret sharing process and take the opportunity to collect the shares from dealer $\mathcal{D}$. Suppose that any adversary acts as participant P_i and attempts to recover the secret. Let the fake participant be P_i' who chooses a private key a_i' ($a_1' \neq a_1$). P_1' computes its public key as $A_1' = a_1'\dot{G}$. However, P_1' cannot publish the public key on SBB since it does not know the credentials of P_i. As a result, during the computation of pseudo share it computes $B_i' = a_i'\dot{A}_0$, $I_i' = g(r, B_i')$, and finally $X_i' = h(x_{i_i'}' \oplus y_{i_i'}')$. The computed X_i' is different from X_i computed by $\mathcal{D}$ since $\mathcal{D}$ computes $B_i == a_i\dot{A}_0 \neq a_i'\dot{A}_0$. $\mathcal{D}$ computes and publishes the verifier $v_i = \{u_i, \Gamma_i\}$ (where $u_i = \gamma_i, h_i.a_0$ and $h_i = H(X_i||PID_i||\Gamma_i)$) for P_i so that anyone

has access to SBB can verify the pseudo share X_i submitted by participant P_i. As a result, the verification of X_i submitted by P_i' fails as $u_i\dot{G} \neq \Gamma_i + h_i'\dot{A}_0$, (since $h_i \neq h_i' = H(X_i'||PID_i||\Gamma_i)$ where $X_i \neq X_i'$).

- *Attack 4: (Collusion attack) Any t or more participants may collude to reveal the secrets.*

  Analysis: Each secret S_i is computed as $S_i = Z_i + W_i^{-1} + \Phi_c^{-1}$, where Z_i is public information with in the group, W_i is computed from the pseudo shares submitted by P_i, and Φ_c is the secret key shared between the dealer and the combiner. As a result, any collusion of t or more participants cannot reveal the secrets.

- *Attack 5: (Combiner cheating) An intruder may impersonate a legal combiner and gain some knowledge about the secrets.*

  Analysis: Any adversary can impersonate as a legitimate combiner in the secret recovery process and take the opportunity to gather the shares from the participants to reveal the secrets. Suppose that any adversary acts as combiner C and attempts to recover the secrets. Let the fake combiner be C' who chooses a private key a_c' ($a_c' \neq a_c$). C' computes its public key as $A_c' = a_c' \cdot G$ ($A_c' \neq A_c$). However, C' cannot publish the public key on SBB since it does not know the credentials of C. As a result, C' creates the verification message $v_i'^{(C)}$ with current timestamp T' as $v_i'^{(C)} = \{a_c' \cdot Ai \cdot H(PID_i||CID||T'), T'\}$ for participant P_i. P_i verify the validity of the combiner by testing $v_i'^{(C)} = a_i \cdot A_c' \cdot H(PID_i||CID||T) \cdot T'$. However, since $a_c \neq a_c'$, $v_i'^{(C)} \neq a_i \cdot A_c \cdot H(PID_i||CID||T) \cdot T$. As a consequence, P_i reject the submission request.

- *Attack 6: An external adversary or any other participant P_j ($i \neq j$) or the dealer or the combiner attempts to reveal the secret shadow of the participant P_i from its public key A_i.*

  Analysis: Each participant $P_i \in \mathcal{P}$ computes its public key as $A_i = a_i \cdot G$. Even if A_i and G are known, computing a_i is hard because solving *ECDLP* is a hard problem in *ECC*. As a result, any external adversary

or participant P_j $(i \neq j)$ or combiner or dealer cannot access the secret shadow (private key) to the participant.

- *Attack 7: An external adversary or any participant P_i or the combiner attempts to reveal the private key of the dealer $\mathcal{D}$ from its public key A_0.*
  Analysis: The dealer $\mathcal{D}$ computes its public key as $A_0 = a_0 \cdot G$. Even if A_0 and G are known, computing a_0 is hard because solving *ECDLP* is a hard problem in *ECC*. As a result, any external adversary or participant or combiner cannot access the private key to the combiner.

- *Attack 8: An external adversary or any participant P_i or the dealer attempts to reveal the private key of the combiner $\mathcal{C}$ from its public key A_c.*
  Analysis: The combiner $\mathcal{C}$ computes its public key as $A_c = a_c \cdot G$. Even if A_c and G are known, computing a_c is hard because solving *ECDLP* is a hard problem in *ECC*. As a result, any external adversary or participant or dealer cannot access the private key to the combiner.

- *Attack 9: A combiner may save one or more pseudo shares submitted by the participants in previous reconstruction processes and use them to compute the secrets in the current process.*
  Analysis: The computation of pseudo shares is dependent on r which is chosen by $\mathcal{D}$ for every sharing process. Therefore, the pseudo shares become invalid after the end of every sharing process. Consequently, the pseudo shares saved by $\mathcal{D}$ are not valid for any other following reconstruction processes.

- *Attack 10: An adversary can retrieve some of the secret keys exchanged between the entities involved in the secret sharing by executing a man-in-the-middle attack.*
  Analysis: The dealer, participants, and combiner must register on the secure bulletin board (SBB), and receive individual credentials (user ID and password) to upload information to SBB. Since all the public keys are uploaded in an authenticated manner to the SBB, it is infeasible to

launch *man-in-the-middle* attack.

- *Attack 11: An intruder attempts to capture some of the submission requests made by a valid combiner and replay those requests (in consecutive sharing processes) to reveal the secrets.*

  Analysis: A combiner has to prove its identity to at least t participants in order to regenerate the original secrets. Without loss of generality, we may assume that an intruder have captured the verification messages $\{v_i^{(c)}\}_{i=1}^{t}$, and resend them to participants $\{P_i\}_{i=1}^{t}$ to impersonate as combiner. After valid verification of intruder as combiner, every $P_i \in \{P_i\}_{i=1}^{t}$ publishes K_i. The intruder use K_i to compute $D_i' = a_c' \cdot A_i$ (since the intruder does not know the private key of $\mathcal{C}$, we can assume $a_c' \neq a_c$). As a result, $D_i' \neq D_i$ and $I_i' = D_i' + K_i \neq I_i'$. Since, the intruder cannot reveal the real pseudo share of the participants, it cannot reveal the secrets.

- Additionally, the proposed scheme provides the following flexibility:

  - *A participant can quit the group, or a new participant may join the group.*

    Analysis: A participant P_{del} can quit the group simply by unregistering from SBB. SBB removes the public key, A_{del} from the SBB, and also remove all the information intended for P_{del} such as v_{del}, Y_{del}, Z_{del}.

    A new participant, P_{new} can join the group by registering with SBB. P_{new} chooses a secret shadow a_{new}, and generates $A_{new} = a_{new} \cdot G$. P_{new} uploads A_{new} in SBB. In the consecutive sharing process, P_{new} and $\mathcal{D}$ exchange the secret key, and $\mathcal{D}$ generates the public share and other public information P_{new} following the alike steps as discussed in Section 4.

  - *A combiner can quit the group, or a new combiner can join the group.*

    Analysis: A combiner C_{del} can quit the group simply by unregistering from SBB. SBB removes the public key, $A_{c_{del}}$ from the SBB.

A new combiner, C_{new} can join the group by registering with SBB. C_{new} chooses a private key $a_{c_{new}}$, and generates $A_{c_{new}} = a_{c_{new}} \cdot G$. C_{new} uploads $A_{c_{new}}$ in SBB. In the consecutive sharing process, P_{new} and $\mathcal{D}$ exchange the secret key. The rest of the process follows the same steps as discussed in Section 4.

- *The secret shadows can be reused (multi-use scheme).*

  Analysis: Each participant $P_i \in \mathcal{P}$ has a secret shadow, a_i and a pseudo share, X_i. X_i is computed from a_i as follows:

$$B_i = a_0 \cdot A_i, \text{ where } A_i = a_i \cdot G,$$
$$I_i = g(r, B_i), \text{ where } I_i \text{ is a point } (x_{I_i}, y_{I_i}) \text{ on } \mathcal{E},$$
$$X_i = H(x_{I_i} \oplus y_{I_i}).$$

  Even if the secret shadow a_i is reused, if the dealer $\mathcal{D}$ selects a new r for each sharing process, the pseudo share will be renewed. Furthermore, the one-way-ness of $g(\cdot,\cdot)$ and $H(\cdot)$ ensures that given r and X_i, it is difficult to find B_i. As a result, reusing a_i for multiple sharing processes is safe.

- *A participant can become a combiner.*

  Analysis: A combiner can be a third party entrusted with the responsibility of reconstructing the secrets for the participants who have submitted valid pseudo shares, or a participant can become a combiner and reconstruct secrets for other participants who collaborate with him. In order to become a combiner, a participant P_i would choose another private key a_c (combiner's private key) along with his own secret shadow a_i (participant's private key), and follow the same steps of a combiner.

5.1. Correctness Analysis

- **Theorem 1: If t or more participants $P_1, P_2, \cdots, P_t$ submit their own shares to the combiner $\mathcal{C}$ (after verifying $\mathcal{C}$), then only secrets can be reconstructed.**

  Proof: The secret reconstruction is possible if at least t pairs of $(x_i, f(x_i)), 1 \leq i \leq t$ are known; otherwise, the reconstruction is failed. Moreover, our proposed scheme is based upon the YCH scheme [28] which is considered as a perfect secret sharing scheme (it guarantees that no knowledge about the secret can be disclosed with less than t shares). Secret reconstruction process is described in the Section 4.6. As a result, in the proposed scheme, none of the secrets can be revealed without having at least t shares from t participants. It ensures the **robustness** of the proposed scheme.

- **Theorem 2: The proposed scheme ensures confidentiality.**

  Proof: If any adversary impersonate as a legitimate participant and tries to collect the shares from dealer $\mathcal{D}$ with an intention to reveal the secrets, that is not possible (as discussed in *Attack 3* of section 5). Also, if any intruder impersonate as a legitimate combiner and try to gather the shares from the participants to reveal the secrets, it is not possible (as discussed in *Attack 5* of section 5). It guarantees the **confidentiality** of the proposed scheme.

- **Theorem 3: The proposed scheme ensures traceability.**

  Proof: Our proposed scheme enables verification of all the entities involved in the secret sharing process: the dealer, the participants, and the combiner. The dealer cannot become a cheater and send incorrect shares to the participants (as discussed in *Attack 1* of section 5); if any participant submits a incorrect share to the combiner, the cheating can be detected and the cheater participant is identified (as discussed in *Attack 2* of section 5); the participants additionally verify a combiner's request for a share contribution (as discussed in 4.4). It ensures the **traceability** of our proposed scheme.

5.2. Comparison with some other works

In this section, the proposed scheme is compared with other existing verifiable secret sharing schemes, which shows the proposed scheme performs comparatively better in a practical configuration where none of the entities can be fully trusted. The comparison is based on the following parameters: (i) ability to share multiple secrets simultaneously; (ii) ability of cheater identification (when cheaters are one or more participants, or the dealer); (iii) ability of combiner verification; (iv) ability of various attack resistance; and (v) use of a secure channel or public channel to share different information among the entities.

The comparison is presented in Table 3. Rajabi and Eslami [42] proposed a verifiable TSS scheme using the collision-resistance and homomorphic properties of generalized compact Knapsack functions. However, the scheme is missing the property of combiner verification. Also, it can share only one secret in a sharing process. Chen et al. [41] presented a verifiable secret sharing scheme, where multiple secrets are distributed through a single sharing method. However, only one secret can be regenerated at each stage (in the proposed scheme, all the secrets can be regenerated in a single stage). Moreover, there is no combiner verification in this scheme. The schemes by Liu et al.[43], Kandar and Dhara [37], Lu et al. [44] and Iwamura et al. [45] are not multi-secret sharing schemes and the schemes [43, 44, 45] lack combiner verification by the participants. Our proposed scheme prevents cheating by the dealer when distributing shares among the participants, whereas the schemes [43, 37, 45] can not guarantee that dealer is not cheating during share distribution and always distribute correct shares.

We compare our proposed scheme with some multi-secret sharing schemes [38, 39, 40]. All these schemes are completely lack the property of verification. They assume all the entities those are involved in the secret sharing process are honest. However, in a practical environment where the entities may not even know each other, none of the entities can be fully trusted. Additionally, we compare our proposed scheme with some other schemes on the basis of their ability to resist various attacks. In the proposed scheme, it is infeasible

to launch any *man-in-the-middle* attack to reveal secret keys shared between the dealer, participants, and combiner. Collusion attacks and replay attacks are also impossible, as we have shown in section 5. However, the schemes in [37, 43, 42, 41, 44, 45] are not resistant to all these attacks, the comparison is as shown in Table 3. Furthermore, the proposed scheme allows the use of public channels to share various types of information between the entities involved in the secret sharing process, whereas the schemes in [43, 42, 40, 39, 44, 45] use some secure channel to exchange information.

Schemes	Year	Multi-secret sharing	Participant verification	Combiner verification	Dealer verification	Resistant to Man in the middle attack	Resistant to collusion attack	Resistant to replay attack	Distribute share over public channel
Liu et al. [43]	2018	yes	yes	no	no	no	yes	no	no
Rajabi-Eslami [42]	2019	no	yes	no	yes	yes	yes	no	no
Chen et al. [41]	2019	yes	yes	no	yes	yes	no	no	yes
Yuan et al. [40]	2020	yes	no	no	no	no	no	no	no
Li et al. [38]	2021	yes	no	no	no	no	no	no	yes
Xu et al. [39]	2021	yes	no	no	no	no	no	no	no
Lu et al. [44]	2022	no	yes	no	yes	yes	yes	no	no
Iwamura et al. [45]	2022	no	yes	no	no	yes	yes	no	no
Kandar and Dhara [37]	2020	no	yes	yes	no	yes	yes	no	no
Our scheme	2023	yes	yes	yes	yes	yes	yes	yes	yes

Table 3: Comparison with other secret sharing schemes

6. Conclusion

In this paper, we proposed a verifiable threshold multi-secret sharing scheme. The proposed scheme can detect cheating by the participants or the combiner and enable the identification of the cheater as well. It prevents any cheating by the dealer. It allows a combiner to quit the group and appoint a new combiner as well. A secure bulletin board (SBB) is used as a trusted authority, and all entities are required to be registered with it. Any entity can access the SBB in an authenticated way, which mitigates the possibility of any *man-in-the-middle* attacks. Furthermore, the proposed scheme does not use a private channel between participants, the dealer, and the combiner for transferring secret keys or pseudo shares. The use of ECC ensures that the scheme is computationally lightweight. The detailed *security analysis* presented for the proposed scheme ensures that the security of the secret shares is dependent on the adversity of solving the hardness of $ECDLP$. In future work, different kinds of possible replay attacks may be studied, and improvements to the scheme can be considered to reduce the quantity of shared information between the entities.

References

[1] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612–613.

[2] G. R. Blakley, Safeguarding cryptographic keys, in: 1979 International Workshop on Managing Requirements Knowledge (MARK), IEEE, 1979, pp. 313–318.

[3] M. Mignotte, How to share a secret, in: Workshop on Cryptography, Springer, 1982, pp. 371–375.

[4] C. Asmuth, J. Bloom, A modular approach to key safeguarding, IEEE transactions on information theory 29 (2) (1983) 208–210.

[5] C.-C. Thien, J.-C. Lin, Secret image sharing, Computers & Graphics 26 (5) (2002) 765–770.

[6] J. Zhao, J. Zhang, R. Zhao, A practical verifiable multi-secret sharing scheme, Computer Standards & Interfaces 29 (1) (2007) 138–141.

[7] M. Ulutas, G. Ulutas, V. V. Nabiyev, Medical image security and EPR hiding using Shamir's secret sharing scheme, Journal of Systems and Software 84 (3) (2011) 341–353.

[8] A. K. Chattopadhyay, A. Nag, K. Majumder, Secure data outsourcing on cloud using secret sharing scheme., IJ Network Security 19 (6) (2017) 912–921.

[9] F. El Mahdi, A. Habbani, Z. Kartit, B. Bouamoud, Optimized scheme to secure IoT systems based on sharing secret in multipath protocol, Wireless Communications and Mobile Computing 2020.

[10] L. Zhang, M. Luo, J. Li, M. H. Au, K.-K. R. Choo, T. Chen, S. Tian, Blockchain based secure data sharing system for Internet of Vehicles: A position paper, Vehicular Communications 16 (2019) 85–93.

[11] Y. Cheng, Z. Fu, B. Yu, Improved visual secret sharing scheme for QR code applications, IEEE Transactions on Information Forensics and Security 13 (9) (2018) 2393–2403.

[12] G. Avoine, S. Vaudenay, Optimistic fair exchange based on publicly verifiable secret sharing, in: Australasian Conference on Information Security and Privacy, Springer, 2004, pp. 74–85.

[13] C.-L. Hsu, T.-S. Wu, Efficient nonrepudiable threshold proxy signature scheme with known signers against the collusion attack, Applied Mathematics and Computation 168 (1) (2005) 305–319.

[14] Y. Liu, Q. Zhao, E-voting scheme using secret sharing and K-anonymity, World Wide Web 22 (4) (2019) 1657–1667.

[15] K. Sutradhar, H. Om, An efficient simulation for quantum secure multiparty computation, Scientific Reports 11 (1) (2021) 1–9.

[16] S. Saha, A. K. Chattopadhyay, S. K. Mal, A. Nag, A secure 'e-tendering'application based on secret image sharing, in: International Conference on Network Security and Blockchain Technology, Springer, 2022, pp. 64–77.

[17] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, Verifynet: Secure and verifiable federated learning, IEEE Transactions on Information Forensics and Security 15 (2019) 911–926.

[18] M. K. Sardar, A. Adhikari, A new lossless secret color image sharing scheme with small shadow size, Journal of Visual Communication and Image Representation (2020) 102768.

[19] Y. Desmedt, S. Hou, J.-J. Quisquater, Audio and optical cryptography, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 1998, pp. 392–404.

[20] J. He, E. Dawson, Multistage secret sharing based on one-way function, Electronics Letters 30 (19) (1994) 1591–1592.

[21] H.-Y. Chien, J.-K. Jan, Y.-M. Tseng, A practical (t, n) multi-secret sharing scheme, IEICE transactions on fundamentals of electronics, communications and computer sciences 83 (12) (2000) 2762–2765.

[22] L. Harn, Comment on" Multistage secret sharing based on one-way function", Electronics Letters 31 (4) (1995) 262.

[23] J. He, E. Dawson, Multisecret-sharing scheme based on one-way function, Electronics Letters 31 (2) (1995) 93–95.

[24] T.-Y. Chang, M.-S. Hwang, W.-P. Yang, A new multi-stage secret sharing scheme using one-way function, ACM SIGOPS Operating Systems Review 39 (1) (2005) 48–55.

[25] H.-X. Li, C.-T. Cheng, L.-J. Pang, An improved multi-stage (t, n)-threshold secret sharing scheme, in: International Conference on Web-Age Information Management, Springer, 2005, pp. 267–274.

[26] A. Endurthi, A. N. Tentu, V. C. Venkaiah, Reusable multi-stage multi-secret sharing scheme based on Asmuth-Bloom sequence, International Journal of Computer Applications 975 (2014) 8887.

[27] A. Basit, N. C. Kumar, V. C. Venkaiah, S. A. Moiz, A. N. Tentu, W. Naik, Multi-stage multi-secret sharing scheme for hierarchical access structure, in: 2017 International Conference on Computing, Communication and Automation (ICCCA), IEEE, 2017, pp. 557–563.

[28] C.-C. Yang, T.-Y. Chang, M.-S. Hwang, A (t, n) multi-secret sharing scheme, Applied Mathematics and Computation 151 (2) (2004) 483–490.

[29] L. Harn, Efficient sharing (broadcasting) of multiple secrets, IEE Proceedings-Computers and Digital Techniques 142 (3) (1995) 237–240.

[30] M. H. Dehkordi, S. Mashhadi, Verifiable secret sharing schemes based on non-homogeneous linear recursions and elliptic curves, Computer Communications 31 (9) (2008) 1777–1784.

[31] C. Hu, X. Liao, X. Cheng, Verifiable multi-secret sharing based on LFSR sequences, Theoretical Computer Science 445 (2012) 52–62.

[32] S. Mashhadi, M. H. Dehkordi, Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem, Information Sciences 294 (2015) 31–40.

[33] A. K. Chattopadhyay, A. Nag, K. Majumder, An ideal multi-secret sharing scheme with verification, Sensors & Transducers 209 (2) (2017) 65.

[34] A. K. Chattopadhyay, P. Maitra, H. N. Saha, A. Nag, A verifiable multi-secret sharing scheme with elliptic curve cryptography, in: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, 2018, pp. 1374–1379.

[35] J. Shao, Efficient verifiable multi-secret sharing scheme based on hash function, Information Sciences 278 (2014) 104–109.

[36] M. Bahramian, K. Eslami, An efficient threshold verifiable multi-secret sharing scheme using generalized Jacobian of elliptic curves, Algebraic Structures and Their Applications 4 (2) (2017) 45–55.

[37] S. Kandar, B. C. Dhara, A verifiable secret sharing scheme with combiner verification and cheater identification, Journal of Information Security and Applications 51 (2020) 102430.

[38] X. Li, C.-C. Chang, Y. Liu, A generalized chinese remainder theorem-based proactive multi-secret sharing scheme for global wide area network, Telecommunication Systems 78 (1) (2021) 49–56.

[39] G. Xu, J. Yuan, G. Xu, Z. Dang, An efficient compartmented secret sharing scheme based on linear homogeneous recurrence relations, Security and Communication Networks 2021.

[40] J. Yuan, G. Xu, G. Xu, An ideal compartmented secret sharing scheme based on linear homogeneous recurrence relations, Cryptology ePrint Archive.

[41] D. Chen, W. Lu, W. Xing, N. Wang, An efficient verifiable threshold multi-secret sharing scheme with different stages, IEEE Access 7 (2019) 107104–107110.

[42] B. Rajabi, Z. Eslami, A verifiable threshold secret sharing scheme based on lattices, Information Sciences 501 (2019) 655–661.

[43] Y. Liu, C. Yang, Y. Wang, L. Zhu, W. Ji, Cheating identifiable secret sharing scheme using symmetric bivariate polynomial, Information Sciences 453 (2018) 21–29.

[44] L. Lu, J. Lu, A lightweight verifiable secret sharing in internet of things, International Journal of Advanced Computer Science and Applications 13 (5).

[45] K. Iwamura, A. A. A. M. Kamal, Ttp-aided secure computation using (k, n) threshold secret sharing with a single computing server, IEEE Access 10 (2022) 120503–120513.

[46] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults, in: 26th Annual Symposium on Foundations of Computer Science (sfcs 1985), IEEE, 1985, pp. 383–395.

[47] L. Harn, C. Lin, Detection and identification of cheaters in (t, n) secret sharing scheme, Designs, Codes and Cryptography 52 (1) (2009) 15–24.

[48] M. Stadler, Publicly verifiable secret sharing, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 1996, pp. 190–199.

[49] J. Shao, Z. Cao, A new efficient (t, n) verifiable multi-secret sharing (VMSS) based on YCH scheme, Applied Mathematics and Computation 168 (1) (2005) 135–140.

[50] R.-J. Hwang, C.-C. Chang, An on-line secret sharing scheme for multi-secrets, Computer Communications 21 (13) (1998) 1170–1176.

[51] M. H. Dehkordi, S. Mashhadi, An efficient threshold verifiable multi-secret sharing, Computer Standards & Interfaces 30 (3) (2008) 187–190.

[52] M. H. Dehkordi, S. Mashhadi, New efficient and practical verifiable multi-secret sharing schemes, Information Sciences 178 (9) (2008) 2262–2274.

[53] M. Hadian Dehkordi, H. Oraei, How to construct a verifiable multi-secret sharing scheme based on graded encoding schemes, IET Information Security 13 (4) (2019) 343–351.

[54] A. Das, A. Adhikari, An efficient multi-use multi-secret sharing scheme based on hash function, Applied mathematics letters 23 (9) (2010) 993–996.

[55] D. R. Stinson, M. Paterson, Cryptography: theory and practice, CRC press, 2018.

[56] A. Antipa, D. Brown, A. Menezes, R. Struik, S. Vanstone, Validation of elliptic curve public keys, in: International Workshop on Public Key Cryptography, Springer, 2003, pp. 211–223.

[57] K.-H. Yeh, A secure IoT-based healthcare system with body sensor networks, IEEE Access 4 (2016) 10288–10299.