

A Code-Based (k, n) -Threshold Secret Sharing Scheme with Integrity Verification

Sapna Jyoti Patel¹ and Sumit Kumar Debnath²

¹National Institute of Technology Jamshedpur, India, 831014, Email:
2023rsma001@nitjsr.ac.in

²Indian Institute of Technology Bhubaneswar, India, 752050, Email:
sumitdebnath@iitbbs.ac.in

Abstract

Threshold secret sharing schemes (TSSS) enable a dealer to distribute a secret among multiple participants such that only authorized subsets can reconstruct the secret while unauthorized subsets obtain no information. Existing *secret sharing schemes (SSS)* are often constrained by limited secret size, non-threshold access structures, or the absence of mechanisms for verifying the authenticity of shares and the integrity of the reconstructed secret. In this paper, we propose a novel (k, n) -threshold secret sharing scheme based on linear *Maximum Distance Separable (MDS)* codes. The proposed construction supports the sharing of comparatively larger secrets by representing the secret as a matrix over a finite field and exploits the linearity of MDS codes to achieve efficient share generation and reconstruction. To strengthen reliability, the scheme incorporates cryptographic hash functions for share authentication and integrity verification of the reconstructed secret. We prove that the proposed scheme satisfies correctness and perfect secrecy, thereby providing *unconditional security* against unauthorized coalitions. Experimental evaluation demonstrates that the proposed construction achieves efficient share generation and reconstruction while outperforming existing code-based secret sharing schemes in terms of supported secret size, scalability, and practical runtime.

Keywords: Threshold Secret Sharing, Code Based Cryptography, MDS Codes, Cryptographic Hash Functions

1 Introduction

Secret sharing [3, 19] is one of the fundamental primitives in modern cryptography for protecting sensitive information against single points of failure. Introduced independently by Shamir [2] and Blakley and Kabatianskii [5], an SSS enables a dealer to distribute a secret among multiple participants such that only authorized subsets can recover the secret, whereas unauthorized subsets obtain no information about it. Among the various secret sharing scheme (SSS) structures, *threshold secret sharing schemes (TSSS)* [19] have attracted considerable attention because of their simplicity and applicability to distributed storage, cloud computing, secure multiparty computation, blockchain systems, and distributed key management.

Although the SSS introduced by Shamir [2] and Blakley and Kabatianskii [5] were developed long before the idea of quantum algorithms was formulated by Shor [8, 9], the development of *Post-Quantum Cryptography (PQC)* [32] has led to the attempts towards designing SSS using quantum-secure primitives. Among the several post-quantum candidates standardized by NIST, code-based cryptography happens to be the one that has withstood both classical and quantum attacks for almost half a decade, making it a promising candidate.

Code-based SSS (CBSSS) [3, 5, 6] constitute an important class of SSS due to their strong algebraic foundation and their close relationship with linear error-correcting codes. Beginning with the work of McEliece and Sarwate [3], and later generalized through Massey's characterization using minimal codewords in [6], several secret sharing constructions have been developed from linear codes. Despite these advances, many existing code-based schemes [11, 14, 15, 18, 20, 22, 23, 27, 29] either do not naturally realize threshold access structures, support only relatively small secrets, or lack mechanisms for

authenticating participant shares and verifying the integrity of the reconstructed secret. Consequently, additional procedures are often required to detect malicious behavior by dishonest participants or a distrustful dealer.

Motivated by these limitations, we propose a code-based (k, n) -TSSS that employs linear MDS codes as its underlying building block. The secret is represented as a matrix over a finite field, enabling substantially larger secrets to be shared within a single execution of the protocol. The proposed construction further incorporates collision-resistant hash functions [16] that allow participants to verify both the authenticity of individual shares and the integrity of the recovered secret without introducing significant computational overhead.

The proposed scheme offers the following contributions.

1. A novel code-based (k, n) -threshold secret sharing scheme using linear MDS codes.
2. Efficient share generation and secret reconstruction algorithms supporting larger secret sizes than existing code-based constructions.
3. End-to-end integrity verification through cryptographic hash functions, enabling authentication of participant shares and verification of the reconstructed secret.
4. Formal proofs establishing correctness, perfect secrecy, and integrity under standard cryptographic assumptions.
5. Experimental evaluation demonstrating favorable runtime and scalability compared with representative code-based secret sharing schemes.

The remaining part of the paper is organized as follows. Section 2 reviews the necessary preliminaries. Section 3 presents the proposed secret sharing scheme. Section 4 analyzes its security. Section 5 evaluates its efficiency and compares it with existing approaches. Finally, Section 6 concludes the paper.

2 Preliminaries

2.1 Entropy Functions [29, 21]

Definition 1 (Entropy Function). *For a random variable X , the entropy function is a measure of the amount of uncertainty (or information) in X . It is given by*

$$H(X) = - \sum_{x \in X} \Pr(X = x) \log [\Pr(X = x)] .$$

Definition 2 (Joint Entropy). *For two random variables X and Y , the joint entropy is a measure of the amount of uncertainty that X and Y jointly carry. Mathematically, the joint entropy of X and Y is*

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} \Pr(X = x, Y = y) \log [\Pr(X = x, Y = y)] .$$

Definition 3 (Conditional Entropy). *For two random variables X and Y , the conditional entropy $H(X|Y)$ is a measure of the amount of uncertainty that X carries, given the condition that the value of Y is already known. Mathematically,*

$$H(X|Y) = H(X, Y) - H(Y) .$$

2.2 Secret Sharing Scheme (SSS) [3, 4, 5, 6]

An SSS is a cryptographic protocol in which a *dealer* $\mathcal{D}$ distributes the shares of a secret S among a set of n participants $\mathcal{P} = \{\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_n\}$ in such a way that only the participants belonging to an authorized subset $\mathcal{A} \subseteq \mathcal{P}$ can successfully recover the secret S , whereas any unauthorized subset $\mathcal{B} \subseteq \mathcal{P}$ obtains no information about it.

Throughout this paper, the term ‘*authorized subset*’ refers to a group of participants that can jointly reconstruct the secret; whereas ‘*unauthorized subset*’ refers to a group that cannot recover any information about the secret, regardless of the shares they collectively possess.

An SSS consists of two algorithms: **Share** and **reconstruct**. The **Share** protocol is used by the dealer $\mathcal{D}$ to generate and distribute the shares $\{s_j\}_{j=1}^n$ among the participants in $\mathcal{P}$. On the other hand, the **Reconstruct** protocol enables an authorized subset $\mathcal{A}$ to recover the secret.

An SSS must possess the following properties:

1. **Correctness:** Any *authorized* subset $\mathcal{A} \subseteq \mathcal{P}$ can reconstruct the secret S correctly using the **Reconstruct** algorithm. Formally, the correctness property of an SSS is established by proving that

$$H(S|S_{\mathcal{A}}) = 0$$

for any authorized subset $\mathcal{A}$.

2. **Secrecy:** No *unauthorized* subset $\mathcal{B} \subseteq \mathcal{P}$ obtains any information about the secret S . In other words, an SSS satisfied the secrecy property if, for any unauthorized subset $\mathcal{B}$, the SSS should have

$$H(S|S_{\mathcal{B}}) = H(S).$$

It is noteworthy that an SSS possessing these two properties is said to have *unconditional security*.

Further, although most of the existing SSS ignore this part, however, it is necessary to ensure that whenever a party reveals shares, the *authenticity* of the shares can be verified. Moreover, the participants of any authorized group of participants should be able to check the *integrity* of the secret so obtained.

Definition 4. *Threshold Secret Sharing Scheme (TSSS) [19] A (k, n) -threshold secret sharing scheme with n participants is the kind of SSS in which any subset of at least k participants form an authorized set, and can successfully recover the original secret.*

2.3 Code-Based Cryptography [12, 13, 21, 26]

Definition 5 (Linear Codes). *Given the values $n, k, d, q \in \mathbb{N}$, with $d, k < n$, a linear code $\mathcal{C}$ with parameters $[n, k, d]_q$ is an injective transformation from $(\mathbb{F}_q)^k$ to $(\mathbb{F}_q)^n$, which can correct upto $\lfloor d/2 \rfloor$ errors that may occur during transmission.*

If $\mathbf{m} \in (\mathbb{F}_q)^k$ is a message vector, then the associated codeword for $\mathbf{m}$ under the code $\mathcal{C}$ is given by $\mathbf{c} = \mathbf{m}\mathbf{G}$, where $\mathbf{G}$ is a full-rank $k \times n$ matrix associated with $\mathcal{C}$, called the *generator matrix*. A notable thing is that, for any generator matrix $\mathbf{G}$ of a code $\mathcal{C}$, at least one set of k columns of $\mathbf{G}$ are linearly independent. Furthermore, the parameters of any code must satisfy the *singleton bound* given by

$$d \leq n - k + 1. \quad (1)$$

2.3.1 MDS Codes and Reed Solomon Codes [3, 13]

In coding theory, *Maximum Distance Separable (MDS) Codes* are a special category of codes in which *any k columns of the generator matrix $\mathbf{G}$ are linearly independent*. This means that any $k \times k$ sub-matrix $\mathbf{G}_k$ formed by taking any k distinct columns of the generator matrix $\mathbf{G}$ of an MDS code will be invertible.

In addition, the parameters of an MDS code are satisfy the equality condition of Equation (1), i.e.,

$$d = n - k + 1.$$

Definition 6 (Reed-Solomon (RS) Codes). *RS codes are a special family of MDS codes with parameters $[q^m - 1, q^m - d, d]$ for any value $d < q^m - 1$.*

3 Proposed Secret Sharing Scheme

We propose a novel TSSS using LEP and linear MDS codes as its fundamental building blocks. We first present a high-level overview of the proposed scheme, followed by a detailed description of each algorithm.

High Level Overview: The proposed SSS consists of three algorithms: the **Init** algorithm, which is run by $\mathcal{D}$ to set the public parameters and the *secret*. The second algorithm, **ShGen**, is executed by $\mathcal{D}$ to generate participant shares for the corresponding *secret*. In order to reconstruct the *secret*, at least k participants must form an authorized coalition and execute the **SecRec** algorithm, which will then give them the original *secret* back. A more detailed explanation of the SSS is given below.

- $(PP, secret) \leftarrow \mathbf{Init}(1^\lambda)$: Using the security parameter λ as input, the dealer $\mathcal{D}$ carries out the following steps:

1. Chooses parameters k, n, q, m in such a way that $n = q^m - 1$ and $k < n$.
2. Chooses a bit string S of length $k^2 \log_2(q^m)$ as the *secret* and sets $r = \left\lceil \frac{|S|}{k \log_2(q^m)} \right\rceil$.
3. Partitions S into blocks of length $\log_2(q^m)$, map each block to an element of $\mathbb{F}_{q^m}$ and arrange the resulting elements row-wise to obtain the matrix $\mathbf{M} \in_R (\mathbb{F}_{q^m})^{r \times k}$. If necessary, append zero-padding to complete the matrix $\mathbf{M}$.
4. Chooses a random generator matrix $\mathbf{G} \in_R (\mathbb{F}_{q^m})^{k \times n}$ of an MDS code, where $\mathbf{G} = [g_{ij}]$ whose columns are $\{G_i\}_{i=1}^n$.

$\mathcal{D}$ sets $PP = (n, k, m, q, r, \mathbf{G}, h)$, where $h : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$ is a cryptographically secure hash function.

- $(\{s_j\}_{j=1}^n, PData) \leftarrow \mathbf{ShGen}(1^\lambda, PP)$: The dealer $\mathcal{D}$ carries out the following steps to generate the participant shares:

1. Chooses a random permutation matrix $\mathbf{P} \in_R (\mathbb{F}_{q^m})^{n \times n}$ and $k-1$ random matrices $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{k-1} \in_R (\mathbb{F}_{q^m})^{n \times n}$. Then, $\mathcal{D}$ sets $\mathbf{P}_k = \mathbf{P} - \sum_{i=1}^{k-1} \mathbf{P}_i$.
2. $\mathcal{D}$ chooses $k-1$ random matrices $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{k-1} \in_R (\mathbb{F}_{q^m})^{r \times n}$ and computes $\mathbf{R}_k = -\sum_{i=1}^{k-1} \mathbf{R}_i$.
3. for $1 \leq j \leq n$, calculates $\mathbf{Q}_j = \sum_{i=1}^k g_{ij} \mathbf{P}_i$ and $\mathbf{S}_j = \sum_{i=1}^k g_{ij} \mathbf{R}_i$.
4. finds the $r \times n$ matrix $\mathbf{G}^*$ as $\mathbf{G}^* = \mathbf{MGP}$.
5. chooses $k-1$ random matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{k-1} \in_R (\mathbb{F}_{q^m})^{r \times n}$ and evaluates $\mathbf{A}_k = \mathbf{G}^* - \sum_{i=1}^{k-1} \mathbf{A}_i$.
6. For $1 \leq j \leq n$, computes $\mathbf{T}_j = \sum_{i=1}^k g_{ij} \mathbf{A}_i$.
7. The share of participant $\mathcal{U}_j$ is then set as $s_j = (\mathbf{X}_j, \mathbf{Q}_j)$, where $\mathbf{X}_j = \mathbf{T}_j + \mathbf{S}_j$.
8. Publishes $PData = (h(\mathbf{M}), h(\mathbf{G}^*), \{j, h(s_j)\}_{j=1}^n)$.

- $\overline{\mathbf{M}} \leftarrow \mathbf{SecRec}(\{\bar{s}_{j_1}, \bar{s}_{j_2}, \dots, \bar{s}_{j_k}\}, PP, PData)$: Consider the set $\mathcal{A} = \{\mathcal{U}_{j_1}, \mathcal{U}_{j_2}, \dots, \mathcal{U}_{j_k}\}$ as the authorized set of k participants with the corresponding shares given by $\{\bar{s}_{j_1}, \bar{s}_{j_2}, \dots, \bar{s}_{j_k}\}$. Let $J = \{j_1, j_2, \dots, j_k\}$. Then, for reconstructing the original secret back, the following steps need to be carried out by the authorized members:

1. Calculate $h(\bar{s}_{j_m})$ for each $m \in A$ and match the results with the corresponding entries in $PData$. If the hash values match, proceed with the reconstruction; else, the affected participants are requested to retransmit their shares.
2. Construct a $k \times k$ matrix $\mathbf{G}_A$ using the j_m^{th} columns of $\mathbf{G}$, where $j_m \in J$.
3. Find the vector $\vec{\mathbf{b}} = [b_1 \ b_2 \ \dots \ b_k]^T$ such that $\vec{\mathbf{b}} = [b_1 \ b_2 \ \dots \ b_k] = \mathbf{G}_A^{-1} [1 \ 1 \ \dots \ 1]_{1 \times k}^T$. Note that, since $\mathbf{G}$ is the generator matrix of an MDS code, any k columns of $\mathbf{G}$ will be linearly independent, and $\mathbf{G}_A$ will be invertible.
4. Parse the shares $\bar{s}_j$ into $\overline{\mathbf{X}}_j$ and $\overline{\mathbf{Q}}_j$.
5. Compute:
 - (a) $\overline{\mathbf{P}} = \sum_{m=1}^k b_m \overline{\mathbf{Q}}_{j_m}$.
 - (b) $\overline{\mathbf{G}}^* = \sum_{m=1}^k b_m \overline{\mathbf{X}}_{j_m}$.
 - (c) $\beta = \overline{\mathbf{G}}^* (\overline{\mathbf{P}}^{-1})$, and hence form β_A by taking all j_m^{th} columns of β .
 - (d) $\overline{\mathbf{M}} = \beta_A (\mathbf{G}_A)^{-1}$.
6. In order to confirm the authenticity and successful recovery of the *secret*, and to make sure that there has been no deception from $\mathcal{D}$'s end while generating and sharing the shares, the participants check whether $h(\overline{\mathbf{M}}) = h(\mathbf{M})$, and $h(\overline{\mathbf{G}}^*) = h(\mathbf{G}^*)$. If both the checks are satisfied, then the recovered *secret* will be the corresponding bit string obtained using the entries of $\overline{\mathbf{M}}$, taken in a row-wise manner.

4 Security Analysis of the Proposed Protocol

According to the security requirements of an SSS introduced in Section 2.2, the proposed TSSS must satisfy the properties of correctness and privacy. In addition, we prove that the scheme provide share authentication and secret integrity. We will prove that the proposed scheme provides unconditional security. The correctness and secrecy properties are established in Section 4.1 and Section 4.2, respectively. Section 4.3 proves share authentication and secret integrity.

4.1 Correctness

Theorem 1. *Under the assumption that there are no errors during transmission and computation processes, any subset of at least k honest participants will always succeed in recovering the original secret in the proposed SSS protocol.*

Proof. Assume that all transmitted shares are received correctly, no computational errors occur, and all participants are honest. Under these assumptions, we have $\bar{s}_{j_m} = s_{j_m}$ for all $j_m \in A$. Thus, we will get

$$(\bar{\mathbf{X}}_{j_m}, \bar{\mathbf{Q}}_{j_m}) = (\mathbf{X}_{j_m}, \mathbf{Q}_{j_m}) . \quad (2)$$

Also, since $\vec{\mathbf{b}} = [b_1 \ b_2 \ \dots \ b_k]^T = \mathbf{G}_A^{-1} [1 \ 1 \ \dots \ 1]_{k \times k}^T$, we will get $\mathbf{G}_A [b_1 \ b_2 \ \dots \ b_k]^T = [1 \ 1 \ \dots \ 1]^T$, i. e.,

$$\begin{bmatrix} g_{1j_1} & g_{1j_2} & \dots & g_{1j_k} \\ g_{2j_1} & g_{2j_2} & \dots & g_{2j_k} \\ \vdots & \vdots & \vdots & \vdots \\ g_{kj_1} & g_{kj_2} & \dots & g_{kj_k} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} .$$

This implies that

$$\sum_{m=1}^k b_m g_{ij_m} = 1 \quad (3)$$

for all i in $1 \leq i \leq k$. Now,

$$\begin{aligned} \bar{\mathbf{P}} &= \sum_{m=1}^k b_m \bar{\mathbf{Q}}_{j_m} \\ &= \sum_{m=1}^k b_m \mathbf{Q}_{j_m} && [\text{Since } \bar{\mathbf{Q}}_{j_m} = \mathbf{Q}_{j_m}] \\ &= \sum_{m=1}^k \left[b_m \left(\sum_{i=1}^k g_{ij_m} \mathbf{P}_i \right) \right] \\ &= \sum_{i=1}^k \left[\left(\sum_{m=1}^k b_m g_{ij_m} \right) \mathbf{P}_i \right] \\ &= \sum_{i=1}^k \mathbf{P}_i && [\text{Using equation (2)}] \\ &= \mathbf{P} && \left[\text{Since } \sum_{i=1}^k \mathbf{P}_i = \mathbf{P} \right] \end{aligned}$$

Similarly,

$$\begin{aligned}
\overline{\mathbf{G}}^* &= \sum_{m=1}^k b_m \mathbf{X}_{j_m} \\
&= \sum_{m=1}^k \left[b_m \left(\sum_{i=1}^k g_{ij_m} \mathbf{A}_i + \sum_{i=1}^k g_{ij_m} \mathbf{R}_i \right) \right] \\
&= \sum_{i=1}^k \left[\left(\sum_{m=1}^k b_m g_{ij_m} \right) \mathbf{A}_i \right] + \sum_{i=1}^k \left[\left(\sum_{m=1}^k b_m g_{ij_m} \right) \mathbf{R}_i \right] \\
&= \sum_{i=1}^k \mathbf{A}_i + \sum_{i=1}^k \mathbf{R}_i \\
&= \mathbf{G}^* + 0 \\
&= \mathbf{G}^*
\end{aligned}
\quad \left[\text{Since } \sum_{i=1}^k \mathbf{A}_i = \mathbf{G}^* \text{ and } \sum_{i=1}^k \mathbf{R}_i = 0 \right]$$

Thus, we have $\overline{\mathbf{P}} = \mathbf{P}$ and $\overline{\mathbf{G}}^* = \mathbf{G}^*$. Therefore, $\beta = \overline{\mathbf{G}}^* (\overline{\mathbf{P}})^{-1} = \mathbf{G}^* \mathbf{P}^{-1} = \mathbf{MGP}(\mathbf{P})^{-1} = \mathbf{MG}$.

Now, note that $\mathbf{MG} = [\mathbf{MG}_1 \ \mathbf{MG}_2 \ \dots \ \mathbf{MG}_n]$. It follows that, if the j^{th} columns of β are taken to form β_A , then we will have $\beta_A = [\mathbf{MG}_{j_1} \ \mathbf{MG}_{j_2} \ \dots \ \mathbf{MG}_{j_k}] = \mathbf{MG}_A$ [because $\mathbf{G}_A$ is formed by taking the j^{th} columns of $\mathbf{G}$, where $j \in A$, giving us $\mathbf{G}_A = [\mathbf{G}_{j_1} \ \mathbf{G}_{j_2} \ \dots \ \mathbf{G}_{j_k}]$].

As a result, the corresponding entries of $\mathbf{M}$ and $\overline{\mathbf{M}}$ will be same and hence, the required checks will always be satisfied and the *secret* will be recovered successfully.

In other words, given $S_A = \{s_j\}_{j \in A}$, the uncertainty about the secret S is zero, i.e.,

$$H(S|S_A) = 0. \quad (4)$$

Hence, the proposed scheme satisfies the *correctness* property. $\square$

4.2 Secrecy

Theorem 2. Any unauthorized coalition $\mathcal{B}$ (with J as the index of the corresponding participants in $\mathcal{B}$) of at most $k-1$ participants will not be able to reconstruct the secret. Moreover, for any two distinct secrets $S, S', s \neq S'$, having respective matrix representations $\mathbf{M}$ and $\mathbf{M}'$, the distribution of shares available to $\mathcal{B}$ is identical, i. e., $\Pr(\{s_j\}_{j \in \mathcal{B}} \mid \text{Secret} = S) = \Pr(\{s_j\}_{j \in \mathcal{B}} \mid \text{Secret} = S')$. Thus, the shares $\{s_j\}_{j \in \mathcal{B}}$ reveal no information about the actual secret. Consequently, the proposed scheme achieves perfect privacy against every unauthorized coalition of at most $k-1$ members.

Proof. Let $J = \{j_1, \dots, j_{k-1}\}$, and let $\mathcal{B} = \{\mathcal{U}_{j_m}\}_{m \in \mathcal{B}}$ be a coalition. Also, let $\mathbf{M}, \mathbf{M}' \in (\mathbb{F}_{q^m})^{r \times k}$, $\mathbf{M} \neq \mathbf{M}'$ correspond to two different secrets S and S' of the same length. Let $\{s_j\}_{j \in \mathcal{B}}$ be the shares constructed using S as the actual secret, and let $\{s'_j\}_{j \in \mathcal{B}}$ be those with S' as the actual secret. We construct a simulator SIM that demonstrates the indistinguishability of the share distribution for the unauthorized subset $\mathcal{B}$ corresponding to any two secrets of equal length.

The simulator is as given in Algorithm 1.

Algorithm 1 The simulator $\mathcal{SIM}$

Input: Public parameters PP , and the index set B .

- 1: Pick an index $j_k \notin B$ to form $B' = B \cup \{j_k\}$ and $\mathcal{B}' = \mathcal{B} \cup \{\mathcal{U}_{j_k}\}$.
 - 2: Form the $k \times k$ matrix $\mathbf{G}_{B'}$ using the j_m^{th} columns of $\mathbf{G}$.
 - 3: Find the vector $[b'_1 \ b'_2 \ \dots \ b'_k]^T$ such that $[b'_1 \ b'_2 \ \dots \ b'_k]^T = \mathbf{G}_{B'}^{-1} [1 \ 1 \ \dots \ 1]_{1 \times k}^T$.
 - 4: For $1 \leq m \leq k-1$, set $\mathbf{Q}'_{j_m} = \mathbf{Q}_{j_m}$ and $\mathbf{X}'_{j_m} = \mathbf{X}_{j_m}$.
 - 5: Set $\mathbf{Q}'_{j_k} = \mathbf{Q}_{j_k}$ and $\mathbf{P}' = \mathbf{P}$ to get $\sum_{m=1}^k b'_m \mathbf{Q}'_{j_m} = \sum_{m=1}^k b'_m \mathbf{Q}_{j_m}$.
 - 6: Compute $(\mathbf{G}^*)' = \mathbf{M}' \mathbf{G} \mathbf{P}' = \mathbf{M}' \mathbf{G} \mathbf{P}$.
 - 7: Set $\mathbf{X}'_{j_k} = \frac{1}{b'_k} \left[(\mathbf{G}^*)' - \sum_{m=1}^{k-1} b'_m \mathbf{X}'_{j_m} \right]$.
 - 8: For any $l \notin B'$, since the columns of $\mathbf{G}_{B'}$ are linearly independent, the l^{th} column can be expressed as a linear combination of the columns of $\mathbf{G}_{B'}$. Find the linear combination and say $G_l = \sum_{m=1}^k c_m^{(l)} G_{j_m}$. For the l^{th} participant, set $\mathbf{Q}'_l = \sum_{m=1}^k c_m^{(l)} \mathbf{Q}'_{j_m}$ and $\mathbf{X}'_l = \sum_{m=1}^k c_m^{(l)} \mathbf{X}'_{j_m}$.
-

Note that, in order to Step 7 to be achievable, we need $b'_k \neq 0$. Since $\mathbf{G}_{B'}$ is a full rank matrix, so from Step 3, we can obtain $b'_k \neq 0$. Thus, we have

1. $\{s'_j\}_{j \in B} = \{(\mathbf{Q}'_j, \mathbf{X}'_j)\}_{j \in B} = \{(\mathbf{Q}_j, \mathbf{X}_j)\}_{j \in B} = \{s_j\}_{j \in B}$.
2. $\sum_{m=1}^k b'_m \mathbf{Q}'_{j_m} = \sum_{m=1}^k b'_m \mathbf{Q}_{j_m}$ (because $\mathbf{P}' = \mathbf{P}$).
3. $\sum_{m=1}^k b'_m \mathbf{X}'_{j_m} = (\mathbf{G}^*)'$.

This will give $\beta = (\mathbf{G}^*)'(\mathbf{P}')^{-1} = (\mathbf{G}^*)\mathbf{P}^{-1} = \mathbf{M}'\mathbf{G}\mathbf{P}(\mathbf{P})^{-1} = \mathbf{M}'\mathbf{G}$, and thus, if $\beta_{B'}$ is the matrix formed by taking the j_m^{th} columns of β , then $\beta_{B'}(\mathbf{G}_{B'})^{-1} = \mathbf{M}'\mathbf{G}_{B'}(\mathbf{G}_{B'})^{-1} = \mathbf{M}'$.

Now, it is noteworthy that, since $\mathbf{X}'_{j_k}$ can be determined for any $\mathbf{M}'$, and since no member of $\mathcal{B}$ has the knowledge about $\mathbf{X}'_{j_k}$ and $\mathbf{Q}'_{j_k}$, it can be concluded that every possible secret of the same length can produce the same shares $\{s_j\}_{j \in B}$ for the members in $\mathcal{B}$.

This proves that

$$\Pr(\{s_j\}_{j \in B} \mid \text{Secret matrix} = \mathbf{M}) = \Pr(\{s_j\}_{j \in B} \mid \text{Secret matrix} = \mathbf{M}') ,$$

which, in turn implies that

$$\Pr(\{s_j\}_{j \in B} \mid \text{secret} = S) = \Pr(\{s_j\}_{j \in B} \mid \text{secret} = S') .$$

Also, this simulation can be generalized for fewer than $k-1$ participants in $\mathcal{B}$. Therefore, the shares observed by any coalition of at most $k-1$ are statistically independent of the secret, implying that the uncertainty about S remains the same even after all the shares of $\mathcal{B}$ are known, i.e.,

$$H(S|S_{\mathcal{B}}) = H(S). \quad (5)$$

Moreover, such a coalition can not reconstruct the secret since the reconstruction algorithm **SecRec** requires k linearly independent columns of the MDS generator matrix $\mathbf{G}$.

Therefore, it can be concluded that the proposed scheme provides *perfect privacy* against any unauthorized coalition. $\square$

4.3 Integrity and Authentication

Theorem 3. Assume that $h : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ is a collision-resistant and cryptographically secure hash function. Then, except with a negligible probability in terms of the security parameter λ ,

1. **Share authenticity:** any modifications in the individual shares s_j is detectable;
2. **Secret integrity:** any alteration in the encoded matrix $\mathbf{G}^*$ or in the recovered secret is detectable.

Consequently, the protocol provides end-to-end integrity from share distribution to secret reconstruction.

Proof. The result follows directly from the collision resistance and preimage resistance of the cryptographic hash function h . In other words,

1. **Individual share authentication:** Each participant can verify the authenticity of the share s_j they received from the dealer $\mathcal{D}$ by calculating $h(s_j)$ and comparing it with the corresponding value published in $PData$. Moreover, during secret reconstruction, the malicious behaviour of a participant, if any, can also be detected in a similar manner. Thus, the protocol ensures that the authenticity of shared can always be verified because any modification to a share changes its hash value and is therefore detectable.
2. **Encoded matrix authentication and secret integrity:** Authenticating $\mathbf{G}^*$ ensures that the encoded matrix being decoded is precisely the one originally shared by the dealer $\mathcal{D}$. Checking whether the values of $h(\mathbf{G}^*)$ and $h(\bar{\mathbf{G}}^*)$ are equal enables the participants of any authorized subset to decide whether there have been any alterations to the encoded matrix. Moreover, in case $h(\mathbf{M}) \neq h(\bar{\mathbf{M}})$. Then, checking whether $h(\mathbf{G}^*) = h(\bar{\mathbf{G}}^*)$ helps in determining whether the encoding was incorrect, the shares were assembled incorrectly, or there have been computation errors. This in turn helps in strengthening the integrity of the secret, because any malicious behaviour of $\mathcal{D}$ can also be easily detected.

Therefore, in the designed protocol, we have that

1. no participant $\mathcal{U}_j$, nor even the dealer $\mathcal{D}$ can tamper with the shares; and
2. $\mathcal{D}$ can not tamper with the encoded matrix or the secret once the corresponding shares have been distributed.

These two properties make the proposed protocol ensure end-to-end integrity. $\square$

5 Parameter Choices and Efficiency Analysis

This section discusses the parameter selection and evaluates the computational efficiency of the proposed scheme. As we have mentioned in Section 3 (Definition 6), we need to use linear MDS codes. Since an important class of MDS codes are the *Reed Solomon codes*, having parameters $[q^m - 1, q^m - d, d]$, we propose the parameter sets for the proposed SSS based on Reed-Solomon codes. The results are as given in Table 1. Further, Figure 1 gives a graphical representation of the **ShGen** and **SecRec** times varying with threshold size. The runtime for different parameter values are as obtained on implementation of the proposed scheme on Dell PowerEdge R640 server by running Ubuntu 20.04.3 LTS (Linux kernel 5.15.0-139-generic). The server was equipped with dual Intel Gold 5218R processors operating at 2.10 GHz, providing 40 physical cores and 256 GB of RAM. For each set of parameters, we used a random $k \times k$ matrix taken as $\mathbf{M}$.

No. of participants (= n)	Threshold (= k)	$(m, q(= 2^m), d)$	Maximum size of secret (= mk^2 bits) (in KB)	ShGen time (in seconds)	SecRec time (in seconds)
15	7	(4, 16, 9)	0.024	0.012	0.003
31	15	(5, 32, 17)	0.137	0.044	0.004
63	33	(6, 64, 31)	0.797	0.208	0.0108
127	65	(7, 128, 63)	3.610	0.976	0.029
255	129	(8, 256, 127)	16.250	5.880	0.100
511	257	(9, 512, 255)	72.563	171.585	1.361

Table 1: Proposed parameter values and corresponding runtimes for the proposed SSS

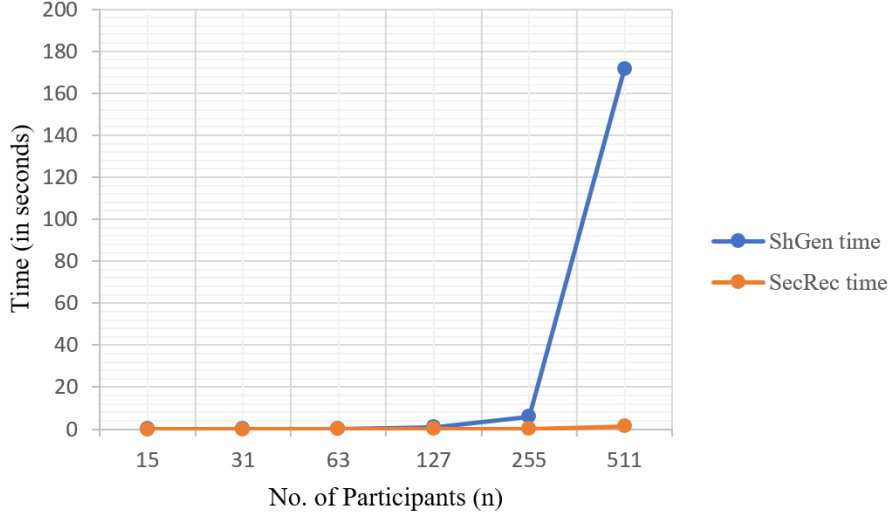


Figure 1: Graph showing variations in runtime for **ShGen** and **SecRec** algorithms of the proposed TSSS

For a better comparison, Table 2 represents some of the existing SSS and their runtimes for a total of 31 participants, with a threshold of 15 members. It is noteworthy here that for the proposed CBSSS with parameters (n, k, m) , the maximum secret size is mk^2 . Thus, for larger parameters, the size of the secret that can be shared with a single execution of the proposed CBGSS increases.

Scheme	Maximum size of secret	ShGen time (in seconds)	Time to find an authorized subset	SecRec time (in seconds)	Expected ShGen time for 12-Byte secret	Expected SecRec time for 12-Byte secret
Blakley and Kabatianskii [5]	1 Byte	0.006267	0.005359	0.232338	0.065	2.788
Massey [6]	1 bit	0.003408	1.004527	0.000020	0.327168	0.00192
Ours	0.137 KB	0.044	0.00	0.004	0.044	0.004

Table 2: Comparison of the proposed SSS with some existing schemes in terms of maximum *Secret* size and different runtimes

Table 2 shows that our scheme outperforms the existing code-based SSS of Blakley and Kabatianskii [5] and Massey [6] on several grounds. The proposed scheme, unlike the existing one, guarantees that every subset containing at least k participants is an authorized set, thereby reducing the time needed to find an authorized set to zero. Moreover, the runtime needed for the **ShGen** and the **SecRec** algorithms for a 12-Byte secret for a total of 31 participants and a threshold of 15 (as given in Table 2) is far less for the proposed scheme when compared to those of the existing ones.

Further, several code-based SSS schemes developed till now (like [11], [15], [18], [20], [23], [27], and [29]) depend heavily on at least one of the two schemes ([5] or [6]), therefore exhibiting performance characteristics similar to either [5] or [6]. This makes our scheme perform better than all of the existing code-based SSS.

Moreover, although the SSS by Zhang, Peng, and Li [25] promises lower runtime, it works only for fewer participants ([25] gives runtime for $n = 11$ participants only). In contrast, the TSSS proposed in this paper is scalable to hundreds of participants, and a higher threshold value also. The maximum secret size supported by a single implementation of the protocol in the proposed TSSS also increases quadratically with the threshold parameter k , making suitable for practical large-scale deployments.

6 Conclusions

This paper presents a novel code-based (k, n) -TSSS constructed using linear MDS codes. Unlike several existing code-based secret sharing schemes, the proposed construction realizes a genuine threshold access structure while simultaneously supporting larger secrets through a matrix-based representation over finite fields. Furthermore, the incorporation of collision-resistant hash functions enables efficient verification of both participant shares and the reconstructed secret, thereby providing end-to-end integrity without significantly increasing computational complexity.

In this paper, it is formally proved that the proposed construction satisfies correctness and achieves perfect secrecy, demonstrating unconditional security against unauthorized coalitions, while also establishing the authenticity of distributed shares and the integrity of the recovered secret. Experimental evaluation indicates that the proposed scheme achieves efficient share generation and reconstruction and compares favorably with representative code-based secret sharing schemes in terms of supported secret size, scalability, and runtime.

References

- [1] Robert J McEliece. “A public-key cryptosystem based on algebraic codes”. In: *Coding Thv* 4244.1978 (1978), pp. 114–116.
- [2] Adi Shamir. “How to share a secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613.
- [3] Robert J. McEliece and Dilip V. Sarwate. “On sharing secrets and Reed-Solomon codes”. In: *Communications of the ACM* 24.9 (1981), pp. 583–584.
- [4] Ernest F Brickell. “Some ideal secret sharing schemes”. In: *Workshop on the Theory and Application of Cryptographic Techniques*. Springer. 1989, pp. 468–475.
- [5] George Robert Blakley and Grigory A Kabatianskii. “Linear algebra approach to secret sharing schemes”. In: *Workshop on Information Protection*. Springer. 1993, pp. 33–40.
- [6] James L Massey. “Minimal codewords and secret sharing”. In: *Proceedings of the 6th joint Swedish-Russian international workshop on information theory*. 1993, pp. 276–279.
- [7] Ed Dawson and Diane Donovan. “The breadth of Shamir’s secret-sharing scheme”. In: *Computers & Security* 13.1 (1994), pp. 69–78.
- [8] Peter W Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.
- [9] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332.
- [10] Oded Goldreich et al. *Foundations of cryptography: volume 1, basic tools*. Vol. 1. Cambridge university press Cambridge, 2001.
- [11] Cunsheng Ding and Jin Yuan. “Covering and secret sharing with linear codes”. In: *International Conference on Discrete Mathematics and Theoretical Computer Science*. Springer. 2003, pp. 11–25.
- [12] San Ling and Chaoping Xing. *Coding theory: a first course*. Cambridge university press, 2004.
- [13] Ron M Roth. *Introduction to coding theory*. Vol. 1. Cambridge University Press Cambridge, 2006.
- [14] Jin Yuan and Cunsheng Ding. “Secret sharing schemes from three classes of linear codes”. In: *IEEE transactions on information theory* 52.1 (2006), pp. 206–212.
- [15] Steven T Dougherty, Sihem Mesnager, and Patrick Sole. “Secret-sharing schemes based on self-dual codes”. In: *2008 IEEE information theory workshop*. IEEE. 2008, pp. 338–342.
- [16] Jonathan J Hoch and Adi Shamir. “On the strength of the concatenated hash combiner when all the hash functions are weak”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2008, pp. 616–630.
- [17] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Vol. 2. Cambridge university press Cambridge, 2009.
- [18] Zhihui Li, Ting Xue, and Hong Lai. “Secret sharing schemes from binary linear codes”. In: *Information Sciences* 180.22 (2010), pp. 4412–4419.

- [19] Noura Al Ebri, Joonsang Baek, and Chan Yeob Yeun. “Study on Secret Sharing Schemes (SSS) and their applications”. In: *2011 International Conference for Internet Technology and Secured Transactions*. IEEE. 2011, pp. 40–45.
- [20] Yun Song and Zhihui Li. “Secret sharing with a class of minimal linear codes”. In: *arXiv preprint arXiv:1202.4058* (2012).
- [21] Mark Kelbert and Yuri Suhov. *Information theory and coding by example*. Cambridge University Press, 2013.
- [22] Selda Çalkavur and Patrick Solé. “Multisecret-sharing schemes and bounded distance decoding of linear codes”. In: *International Journal of Computer Mathematics* 94.1 (2017), pp. 107–114.
- [23] Jon-Lark Kim and Nari Lee. “Secret sharing schemes based on additive codes over GF (4)”. In: *Applicable Algebra in Engineering, Communication and Computing* 28.1 (2017), pp. 79–97.
- [24] Nicolas Sendrier. “Code-based cryptography: State of the art and perspectives”. In: *IEEE Security & Privacy* 15.4 (2017), pp. 44–50.
- [25] En Zhang, Jie Peng, and Ming Li. “Outsourcing secret sharing scheme based on homomorphism encryption”. In: *IET Information Security* 12.1 (2018), pp. 94–99.
- [26] Harshdeep Singh. “Code based cryptography: Classic mceliece”. In: *arXiv preprint arXiv:1907.12754* (2019).
- [27] Adel Alahmadi et al. “A multisecret-sharing scheme based on LCD codes”. In: *Mathematics* 8.2 (2020), p. 272.
- [28] Jean-François Biasse et al. “LESS is more: code-based signatures without syndromes”. In: *International Conference on Cryptology in Africa*. Springer. 2020, pp. 45–65.
- [29] Selda Çalkavur et al. *Code based secret sharing schemes: applied combinatorial coding theory*. World Scientific, 2022.
- [30] Arup Kumar Chattopadhyay et al. “Secret sharing: A comprehensive survey, taxonomy and applications”. In: *Computer Science Review* 51 (2024), p. 100608.
- [31] Jingyu Chen et al. “Lattice-based threshold secret sharing scheme and its applications: A survey”. In: *Electronics* 13.2 (2024), p. 287.
- [32] NIST PQC Team. *What Is Post-Quantum Cryptography?* <https://www.nist.gov/cybersecurity-and-privacy/what-post-quantum-cryptography>. 2024.