MICODE: A Minimal Code Design for Secret Sharing Scheme

Belkacem Imine[1][a], Rahul Saha[2,3] and Mauro Conti[2]
[1]*LACOSI Laboratory, Faculty of Electrical Engineering, University of Science and Technology of Oran Mohamed Boudiaf (USTO-MB), Oran, Algeria*
[2]*Department of Mathematics, University of Padua, Padua, Italy*
[3]*School of Computer Science and Engineering, Lovely Professional University, Punjab, India*

Keywords: Reed-Muller, Code, Secret, Sharing, Minimal.

Abstract: Secret Sharing Schemes (SSS) in cryptography often utilize minimal linear codes for efficiency, with minimum distance playing a crucial role. Determining minimal codewords in general linear codes presents a challenge known as the linear code covering problem. To address this, we propose MInimal COde DEsign (*MICODE*), a novel method for generating minimal codes from binary Reed-Muller (RM) codes using the Ashikhmin-Barg lemma. Unlike existing approaches limited to small RM codes, MICODE extends to higher orders through a systematic puncturing strategy. By recursively removing one-weight columns from the generator matrix, we reduce the RM code's maximum Hamming weight preserving its minimum distance. The RM generator matrix's structure, derived from the Kronecker product of lower-triangular binary matrices, facilitates this construction. We conduct rigorous mathematical analysis of MICODE establishing parameters for a secure SSS. While these minimal codes are unsuitable for error correction due to their reduced code rate, they are proven highly effective for cryptographic applications, such as Massey SSS, where security depends on minimum distance. Our analysis also explores trade-offs between code rate and error performance offering new insights into their theoretical and practical implications.

1 INTRODUCTION

Confidentiality is a fundamental service of cryptography, achieved through symmetric and asymmetric methods. However, the security of these methods relies heavily on the protection of secret keys. According to Kerckhoff's principle, the security of a cryptosystem depends entirely on the secrecy of its encryption key. Storing the key in a single device poses a risk of system failure, while storing multiple copies across devices increases the risk of exposure. To address these challenges, Secret Sharing Schemes provide a robust cryptographic solution, ensuring both the security and availability of secret keys by distributing them across multiple parties without compromising confidentiality. Blakley (Blakley, 1979) and Shamir (Shamir, 1979) schemes are the first proposed secret sharing schemes. Shamir's SSS is a (k,n)-threshold SSS with $\mathcal{P}$ as the dealer and n participants $P = \{p_1, ..., p_n\}$. Every participant p_i keeps one share and any group of k or more can figure out the secret s. The number of authorized participants who can recover the secret s in this scheme is at least k. The set containing these authorized participants is called an access set, whereas the set containing the smallest number of authorized participants is called a minimal access set and has a size of k. Shamir's SSS is much more popular because all minimal access sets are of the same size, which makes the implementation easier.

In (McEliece and Sarwate, 1981), the researchers suggested employing Error Correction Codes (ECC_s) to construct a SSS, wherein a (k,n)-threshold scheme is generalized using the Reed-Solomon code. Massey has proposed an ECC-based SSS with different-sized minimal access sets (Massey, 1993), where the access structure is defined by the minimal codewords of the dual code. A nonzero codeword in a linear code is considered minimal if its support is not a proper superset of the support of any other nonzero codeword. The general properties of minimal codewords were discussed in (Agrell, 1996; Agrell, 1998). The Massey's secret sharing involves determining the

[a] https://orcid.org/0000-0002-2295-5145

Imine, B., Saha, R. and Conti, M.
MICODE: A Minimal Code Design for Secret Sharing Scheme.
DOI: 10.5220/0013517300003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 381-391
ISBN: 978-989-758-760-3; ISSN: 2184-7711

minimal access sets by computing all the minimal codewords in a linear code. However, we observe in the literature that minimal codewords are difficult to determine (Covering problem) (Ding and Yuan, 2003; Li et al., 2010); thus, Massey's secret sharing is difficult to implement. This invokes the search for a class of codes in which determining minimal codewords is easier, or in other words, in which solving the covering problem is simple. For specific families of linear codes, minimal codewords have been determined by leveraging their structural properties (Ding et al., 2000; Schillewaert et al., 2010; Yuan and Ding, 2005). Reed-Muller codes are an intriguing class of codes. Still, the covering problem has only been solved for second-order binary Reed-Muller codes (Ashikhmin and Barg, 1998), or only minimal low-weight codewords have been determined in Reed-Muller codes(Schillewaert et al., 2010). As a result, this leads to the search for a class of codes in which all codewords are minimal to facilitate determining the access structure of the secret sharing scheme. This class is called minimal linear code.

Ashikhmin and Barg propose a sufficient condition for determining whether or not a linear code is minimal (Ashikhmin and Barg, 1998). Yun Song and Zhihui Li have proposed a minimal linear code construction based on irreducible cyclic codes and have investigated the conditions under which a given irreducible cyclic code is minimal (Song and Li, 2012). In (Xu and Qu, 2019), the authors construct three classes of minimal linear codes over finite fields. Even if the Ashikhmin-Barg inequality is not satisfied, another method allows for the construction of a minimal code (Chang and Hyun, 2018; Ding et al., 2018). On the other hand, Chang and Hyun have presented a method for constructing a class of minimal binary linear codes that violate the Ashikhmin-Barg inequality by using Boolean functions derived from mathematical objects known as simplicial complexes. Another necessary and sufficient condition presented in (Ding et al., 2018) allows for the generation of three infinite classes of minimal binary linear codes, which are based on the Walsh-Hadamard transform of a Boolean function. The work also demonstrates that these three classes of minimal binary linear codes violate the Ashikhmin-Barg inequality.

1.1 Motivations and Contributions

Our proposed MICODE has the following contributions.

- **Minimal Code Construction with a Large Minimum Distance:** Our proposed minimal code (MICODE) constructs a minimal linear code with a large minimum distance, a property essential for secret sharing schemes (SSS). Unlike Ding's minimal code (Ding et al., 2018), MICODE achieves a higher minimum distance, enhancing both the security and robustness of SSS applications. Our approach leverages the structured properties of RM codes, where the minimum distance is well-defined. Given the generator matrix of an RM code, we recursively reduce the maximum Hamming weight by applying a puncturing scheme to one-weight columns and their corresponding rows until the Ashikhmin-Barg inequality is satisfied. This ensures that the resulting code remains minimal while maintaining the same minimum distance as the original RM code
- **Ideal and Perfect SSS:** Our contribution presents a perfect and ideal code-based secret sharing scheme based on the dual of the generated minimal code. The scheme ensures that only authorized participants can reconstruct the secret, while unauthorized sets learn nothing. It is efficient and easy to implement, as all minimal access sets have the same size, similar to Shamir's SSS (Shamir, 1979). However, our approach offers significant advantages over Shamir's SSS, including fine-grained access control, enhanced security, customizable access structures, and resilience to share compromise, making it particularly suitable for applications requiring specific access rules and higher security.

1.2 Paper Organization

We organize the rest of the paper in the following sections. Section 2 provides general definitions concerning linear codes, how to construct a Reed-Muller code, and minimal code characteristics. Section 3 introduces MICODE and explains the steps of generating a minimal code from the Reed-Muller code. Section 4 shows an application of MICODE in SSS. It describes a perfect Massey's secret sharing based on the generated minimal code and the number of possible minimal access sets. The last section of the paper concludes the paper.

2 AN OVERVIEW OF REED-MULLER CODES AND MINIMAL CODES

Let $\mathbb{F}_q$ be a finite field of q elements. An $[n,k]_q$ $q-$ary code C is a $k-$dimensional subspace of the vector

space $\mathbb{F}_q^n$ such that the sum of any two codewords $u, v \in C$ produces another codeword $c = u + v \in C$.

Definition 1 (Hamming weight)**.** *Let $v = (v_0, ..., v_{n-1}) \in \mathbb{F}_q^n$ be a vector, then the Hamming weight $w_H(v)$ of v is the number of its non-zero coordinates.*

$$w_H(v) \triangleq |\{i \mid v_i \neq 0\}|. \quad (1)$$

Definition 2 (Hamming distance)**.** *The Hamming distance d_H between two vectors $u, v \in \mathbb{F}_q^n$ is defined by:*

$$d_H(u,v) \triangleq |\{i \mid u_i \neq v_i\}|. \quad (2)$$

A $[n,k,d]$ code C is a linear code in which the Hamming distance between any two distinct codewords is at least d.

Definition 3 (Support of a vector)**.** *The support of a vector $v \in \mathbb{F}_q^n$ is the set of indices of its nonzero coordinates.*

$$Supp(v) \triangleq \{i \mid v_i \neq 0, \ 0 \leq i \leq n-1\}. \quad (3)$$

Let C be a linear code, we say that a codeword $c \in C$ covers a codeword $c' \in C$ if and only if $Supp(c') \subseteq Supp(c)$.

Definition 4 (Evaluation function)**.** *Consider the polynomial ring $F_p = [x_0, ..., x_{p-1}]/(x_0^2 - x_0, ..., x_{p-1}^2 - x_{p-1})$ with p variables, and $\mathbb{F}_2 = \{0,1\}$ is a binary field. The elements of $\mathbb{F}_2^p$ are ordered in decreasing index order. Let $n = 2^p$, the evaluation function of a monomial $g \in F_p$ is the evaluation function $Eval(g)$ at all n vectors of $\mathbb{F}_2^p$.*

$$Eval: \ g \in F_p \rightarrow (g(v))_{v \in \mathbb{F}_2^p}. \quad (4)$$

Example 1. *Table 1 displays the evaluation of all the monomials with 3 variables and degree less than or equal to 2.*

Table 1: The evaluation of all monomials with 3 variables and degree less than or equal to 2 (Abbe et al., 2020).

$Eval(\mathbf{1})$	1	1	1	1	1	1	1	1
$Eval(x_1x_2)$	1	1	0	0	0	0	0	0
$Eval(x_0x_2)$	1	0	1	0	0	0	0	0
$Eval(x_0x_1)$	1	0	0	0	1	0	0	0
$Eval(x_2)$	1	1	1	1	0	0	0	0
$Eval(x_1)$	1	1	0	0	1	1	0	0
$Eval(x_0)$	1	0	1	0	1	0	1	0

Definition 5 (Reed-Muller code)**.** *The r-th order binary Reed-Muller code $RM(p,r)$ code is defined by evaluating all $p-$variate monomials with degree less than or equal to r.*

$$RM(p,r) \triangleq \{Eval(g) \mid g \in F_p, \ deg(g) \leq r\}. \quad (5)$$

The code parameters of an $RM(p,r)$ code are as follows:

- *Code length:* 2^p.
- *Code dimension:* $\sum_{i=0}^{r} \binom{p}{i}$.
- *Minimum distance:* 2^{p-r}.

Example 2. *The $RM(3,1)$ is defined by the evaluation of all $3-$variate monomials of degree less than or equal to 1 (Abbe et al., 2020).*

$$G_{RM(3,1)} = \begin{bmatrix} Eval(x_2) \\ Eval(x_1) \\ Eval(x_0) \\ Eval(\mathbf{1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Definition 6 (Minimal codeword (Ding and Salomaa, 2006))**.** *Let $[n,k]_q$ be a linear code over $\mathbb{F}_q$. A non-zero codeword $c \in C$ is minimal if it does not cover any other codeword $c' \in C$.*

Definition 7 (Minimal code (Ding and Salomaa, 2006))**.** *A code C is minimal if all of its non-zero codewords are minimal.*

Lemma 1 (Ashikhmin-Barg (Ashikhmin and Barg, 1998))**.** *An $[n,k,d]_q$ code C is minimal if:*

$$\frac{W_{min}}{W_{max}} > \frac{q-1}{q}. \quad (6)$$

W_{min} and W_{max} denote the non-zero minimum-weight codeword and the miximum-weight codeword in C, respectively.

3 OUR PROPOSAL: MICODE

In this section, we introduce our algorithm for constructing a minimal code from the Reed-Muller code. We use the upper bound described in (Ball and Blokhuis, 2013) to compute the maximum weight W_{max} for a linear code such that the Hamming distance is preserved in the resulting minimal code.

Theorem 1. *Let C be an $[n,k,d]_q$ code such that q is prime, then the maximum weight W_{max} fulfils (Ball and Blokhuis, 2013)*

$$W_{max} \leq (n-d)q - e(q-1), \quad (7)$$

where $e \in \{0,1,...,k-2\}$ is the maximal integer satisfying:

$$\binom{n-d}{e} \not\equiv 0 \ (mod \ q^{k-1-e}). \quad (8)$$

Our algorithm is a recursive algorithm that produces a minimal code from a binary *RM* code of length n, dimension k, and minimum distance d. It takes a generator matrix G at each instance and verifies the Ashikhmin-Barg condition to ensure minimality, such that the Hamming distance remains unchanged. At each instance for a given G, our algorithm searches for all columns with Hamming weights of one, then:

- If the computed columns are all different: it deletes one column and one row in such a way that the deleted row has the greatest Hamming weight and the intersection of the deleted row and column is one.
- If at least two columns match: it deletes all matching columns as well as the corresponding row, which is indexed by the position of their non-zero coordinates.
- This process is repeated until the generator matrix satisfies the Ashikhmin-Barg condition for minimality, specifically $\frac{d}{W_{max}} > \frac{1}{2}$.

Algorithm 1 describes in detail how to generate minimal code from a *RM* code.

Example 3. *Suppose the* $RM(4,2)$ *code generated by the generator matrix* $G_{RM(4,2)}$.

$$G_{RM(4,2)} = \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0 \\ 1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

- *Step 1: There is only one column of Hamming weight 1, we remove the last column (column 15) and the last row (row 10).*

$$G_1 = \begin{bmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&1&1&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0&1&1&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0&1&0&1&0&0&0&0 \\ 1&1&1&1&0&0&0&0&1&1&1&1&0&0&0 \\ 1&0&0&0&1&0&0&0&1&0&0&0&1&0&0 \\ 1&1&0&0&1&1&0&0&1&1&0&0&1&1&0 \\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ \hline 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

- *Step 2: Following (Ball and Blokhuis, 2013), we compute* W_{max} *of the code generated by* G_1. *Given* $n_1 = 15, k_1 = 10$, *and* $d = 4$, *we obtain* $W_{max} \leq 14$ *for* $e = 8$.
- *Step 3: We compute* $(\frac{d}{Min(n_1, W_{max})}) = \frac{4}{14} = 0.2857 < \frac{1}{2}$. *We proceed to the next step.*
- *Step 4: The columns* $\{7,11,13,14\}$ *have Hamming weight* 1, *then* $I = \{7,11,13,14\}$ *and* $T = \{Supp(g_7), Supp(g_{11}), Supp(g_{13}), Supp(g_{14})\} = \{3,6,8,9\}$. *Since all the rows indexed by T have the same Hamming weight, we can remove either (row* 3 *and column* 7*), (row* 6 *and column* 11*), (row* 8 *and column* 13*), or (row* 9 *and column* 14*).*

Input: Generator matrix G of $[n,k,d]$-RM code C
Output: A generator matrix G_m of a minimal code
Function `Puncture(G)`**:**
- $n \leftarrow$ Number of columns of G ;
- $k \leftarrow$ Number of rows of G ;
- $n' \leftarrow n$;
- $k' \leftarrow k$;
- $W_{max} \leftarrow n$;
- $I \leftarrow \emptyset$;
- $T \leftarrow \emptyset$;
- **for** $j \leftarrow 0$ *to* $n-1$ **do**
  - **if** $w_H(g_j) = 1$ **then**
    - $I \leftarrow I \cup \{j\}$;
  - **end**
- **end**
- **for** $i \leftarrow 0$ *to* $k-1$ **do**
  - $T \leftarrow \{T \cup i \mid G_{i,j} = 1, \forall j \in I\}$;
- **end**
- $R(T) \leftarrow$ The rows of G indexed by T;
- $\hbar^\ell \subseteq I \leftarrow \{x_1, \ldots, x_m \mid Supp(g_{x_1}) = \cdots = Supp(g_{x_m}) = \ell, 0 \leq x_1 < x_2 < \ldots \leq m < n\}$;
- **if** $|\hbar^\ell| > 1$ **then**
  - Delete the row indexed by ℓ from G;
  - Delete the columns indexed by $\hbar^\ell$ from G;
  - $n' \leftarrow n - |\hbar^\ell|$;
  - $k' \leftarrow k-1$;
  - $W_{max} \leftarrow Max\_weight([n',k',d])$ (Ball and Blokhuis, 2013) ;
- **else**
  - $r_{max} \leftarrow r \in R(T) \mid w_H(r) \geq r', \forall r' \in R(T)$;
  - **for** $j \leftarrow 0$ *to* $n-1$ **do**
    - $g' \leftarrow \{g_j \mid r_{max} \cap g_j = 1, j \in I\}$;
  - **end**
  - Delete the row r_{max} from G;
  - Delete the column g' from G;
  - $n' \leftarrow n-1$;
  - $k' \leftarrow k-1$;
  - $W_{max} \leftarrow Max\_weight([n',k',d])$ (Ball and Blokhuis, 2013);
- **end**
- **if** $(d/(Min(W_{max}, n')) > \frac{1}{2})$ **then**
  - **return** $G_m = G$
- **else**
  - Puncture(G)
- **end**

End Function

Algorithm 1: Constructing Minimal code from RM(p,r).

Let us take the first case.

$$G_2 = \left[\begin{array}{cccccccc|cccccccc} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0\\ \hline 1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\ 1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\ 1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\ 1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&1 \end{array}\right]$$

- *Return to steps 2 and 3, $W_{max} \leq 14$ for $n_2 = 14, k_2 = 9$, and $d = 4$. We compute $(\frac{d}{Min(n_2, W_{max})}) = \frac{4}{14} = 0.2857 < \frac{1}{2}$. We proceed to the next step.*
- *Step 5: $I = \{10,12,13\}$ and $T = \{Supp(g_{10}), Supp(g_{12}), Supp(g_{13})\} = \{5,7,8\}$. Since all the rows indexed by T have the same Hamming weight, we can remove either (row 5 and column 10), (row 7 and column 12), or (row 8 and column 13). Let us take the first case.*

$$G_3 = \left[\begin{array}{cccccccccc|cccc} 1&1&1&1&0&0&0&0&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&1&1&0&0&0&0&0\\ 1&0&1&0&0&0&0&1&0&1&0&0&0&0\\ \hline 1&1&1&1&0&0&0&1&1&1&1&0&0&0\\ 1&0&0&0&1&0&0&1&0&0&0&1&0&0\\ 1&1&0&0&1&1&0&1&1&0&0&1&1&0\\ 1&0&1&0&1&0&1&1&0&1&0&1&0&1 \end{array}\right]$$

- *Return to steps 2 and 3, $W_{max} \leq 13$ for $n_3 = 13, k_3 = 8$, and $d = 4$. We compute $(\frac{d}{Min(n_3, W_{max})}) = \frac{4}{13} = 0.3077 < \frac{1}{2}$. We proceed to the next step.*
- *Step 6: $I = \{3,11,12\}$ and $T = \{Supp(g_3), Supp(g_{11}), Supp(g_{12})\} = \{0,6,7\}$. Since the row 0 has a weight 4 and rows $\{6,7\}$ have weights 8, we can remove either (row 6 and column 11) or (row 7 and column 12). Let us remove (row 6 and column 11).*

$$G_4 = \left[\begin{array}{cccccccccccc|c} 1&1&1&1&0&0&0&0&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&1&1&0&0&0&0&0\\ 1&0&1&0&0&0&0&1&0&1&0&0&0&0\\ 1&0&0&0&1&0&0&1&0&0&1&0&0&0\\ \hline 1&1&0&0&1&1&0&1&1&0&1&1&0&0\\ 1&0&1&0&1&0&1&1&0&1&1&0&1&1 \end{array}\right]$$

- *Return to steps 2 and 3, $W_{max} \leq 12$ for $n_4 = 12, k_4 = 7$, and $d = 4$. We compute $(\frac{d}{Min(n_4, W_{max})}) = \frac{4}{12} = 0.3333 < \frac{1}{2}$. We proceed to the next step.*
- *Step 7: $I = \{3,5,8,11\}$ and $T = \{0,1,3,6\}$. The rows $\{0,1,3\}$ have Hamming weight 4 and the row 6 has Hamming weight 8. We must remove (row 6 and column 11), we obtain:*

$$G_5 = \left[\begin{array}{cccccccccccc|c} 1&1&1&1&0&0&0&0&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&0&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&1&1&0&0&0&0&0\\ 1&0&1&0&0&0&0&1&0&1&0&0&0&0\\ 1&0&0&0&1&0&0&1&0&0&1&0&0&0\\ \hline 1&0&1&0&1&0&1&1&0&1&1&0&1&1 \end{array}\right]$$

- *Return to steps 2 and 3, $W_{max} \leq 10$ for $n_5 = 11, k_5 = 6$, and $d = 4$. We compute $(\frac{d}{Min(n_5, W_{max})}) = \frac{4}{10} = 0.4 < \frac{1}{2}$. We proceed to the next step.*
- *Step 8: $I = \{3,5,6,8,9,10\}$ and $T = \{0,1,2,3,4,5\}$. All rows indexed to T have Hamming weight 4. We can remove either (row 0 and column 3), (row 1 and column 5), (row 2 and column 6), (row 3 and column 8), (row 4 and column 9), or (row 5 and column 10). Let us remove (row 0 and column 3), then we obtain:*

$$G_6 = \left[\begin{array}{ccccccccccc|c} 1&1&1&0&0&0&0&0&0&0&0&0\\ \hline 1&1&0&0&1&1&0&0&0&0&0&0\\ 1&0&1&0&1&0&1&0&0&0&0&0\\ 1&1&0&0&0&0&0&1&1&0&0&0\\ 1&0&1&0&0&0&0&1&0&1&0&0\\ 1&0&0&0&1&0&0&1&0&0&1&0 \end{array}\right]$$

- *Return to steps 2 and 3, $W_{max} \leq 10$ for $n_6 = 10, k_6 = 5$, and $d = 4$. We compute $(\frac{d}{Min(n_6, W_{max})}) = \frac{4}{10} = 0.4 < \frac{1}{2}$. We proceed to the next step.*
- *Step 9: $I = \{4,5,7,8,9\}$ and $T = \{0,1,2,3,4\}$. All rows indexed to T have Hamming weight 4. We can remove either (row 0 and column 4), (row 1 and column 5), (row 2 and column 7), (row 3 and column 8), or (row 4 and column 9). Let us remove (row 0 and column 4), then we obtain:*

$$G_7 = \left[\begin{array}{cccccccccc|c} 1&1&0&1&1&0&0&0&0&0&0\\ \hline 1&0&1&1&0&1&0&0&0&0&0\\ 1&1&0&0&0&0&1&1&0&0&0\\ 1&0&1&0&0&0&1&0&1&0&0\\ 1&0&0&1&0&0&1&0&0&1&0 \end{array}\right]$$

- *Return to steps 2 and 3, $W_{max} \leq 9$ for $n_7 = 9, k_7 = 4$, and $d = 4$. We compute $(\frac{d}{Min(n_7, W_{max})}) = \frac{4}{9} = 0.444 < \frac{1}{2}$. We proceed to the next step.*

- *Step 10:* $I = \{1,4,6,7,8\}$ *and* $T = \{0,1,2,3\}$. *Two columns are identical (*1 *and* 6*) and have* 1 *in position* 1*, so* $\hbar^1 = \{1,6\}$. *We must remove (row* 1*, column* 1 *and column* 6*), then we obtain:*

$$G_8 = \left[\begin{array}{c|c|c|c|c|c|c|c|c} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

- *Return to steps 2 and 3,* $W_{max} \leq 5$ *for* $n_8 = 7, k_8 = 3$*, and* $d = 4$. *We compute* $(\frac{d}{Min(n_8, W_{max})}) = \frac{4}{5} = 0.8 > \frac{1}{2}$. *Therefore,* G_8 *is a valid generator matrix of a minimal code of length* $n = 7$*, dimension* $k = 3$*, and minimum distance* $d = 4$.

Proposition 1. *For any* $RM(p,r)$ *code of length n, dimension k and minimum distance d there exists an minimal code of length* $n' \leq n$*, dimension* $k' \leq k$ *and minimum distance d such that:*

- *If* $p = r$*, then* $n' = k' = 1$.
- *If* $r = 0$ *and* $p > 0$*, then* $k = k' = 1$ *and* $n' = n$.
- *If* $0 < r < p$*, then* $k' \geq 2$ *and* $n' < n$.

Proof. If $p = r$, the $RM(p,r)$ code's generator matrix G is formed by the evaluation vectors of all the $p-$variate monomials, i.e. $n = k$, implying the existence of the row $(1,0,...,0)$ in the generator matrix G and that its support is covered by all the other $k-1$ rows. As a result, all of the remaining $k-1$ rows and $n-1$ columns $(g_1, g_2, ...g_{n-1})$ must be deleted. If $r = 0$ and $p > 0$, then the $RM(p,r)$ has only one nonzero codeword which is minimal, so $k = k' = 1$ and $n' = n$.
If $0 < r < p$, then the $RM(p,r)$ code contains at least two codewords of weight $d = 2^{p-r}$. Let $G^{(i)}$ be the matrix produced by our algorithm at instance i, if the removed row from $G^{(i)}$ has a weight d, then no codeword of a weight greater than d exists. If $G^{(i)}$ has two rows, they must have the same weight and have "1" in the first position, this means $\frac{d}{W_{max}^{(i)}} > \frac{1}{2}$. □

Proposition 2 (Maximum number of iterations)**.** *Let* C *be an* $[n,k,d]_2$ $RM(p,r)$ *code,* $0 < r < p$ *and* C' *the generated* $[n',k',d]_2$ *minimal code. The maximum number of iterations* I_{max} *required to move from code* C *to code* C' *is the smallest integer i that satisfies:*

$$n - \frac{e'}{2} - 2d < i, \;\; 0 \leq i \leq k-2 \tag{9}$$

Such that $e' \in \{0,...,k-i-2\}$ *is the maximum integer that satisfies:*

$$\binom{n-i-d}{e'} \not\equiv 0 \; (mod\; 2^{k-i-1-e'}). \tag{10}$$

Proof. Let W'_{max} be the maximum Hamming weight in C'. According to equations 6 and 7 we have:

$$W'_{max} \leq 2(n'-d) - e' < 2d$$

Such that $e \in \{0,...,k'-2\}$ is the maximum integer that satisfies $\binom{n'-d}{e'} \not\equiv 0 \pmod{2^{k'-1-e'}}$. In the worst-case scenario, our algorithm allows us to delete only one column and one row for each instance where condition 6 is not met. Assume that the produced code is minimal after i iterations, which means that $n' = (n-i)$ and $k' = (k-i)$ resulting in the following:

$$\begin{aligned} 2(n-i-d) - e' &< 2d \\ 2n - 2i - 4d - e' &< 0 \\ n - 2d - \frac{e'}{2} &< i \end{aligned}$$

Because $0 < r < p$, the dimension of the minimal code produced must be at least two. As a result, the smallest integer i that solves the preceding equation is the maximum number of iterations. □

Definition 8. *A* $[n,k,d]$*-binary simplex code is a linear code characterized by the following properties (Helleseth et al., 2004):*

- *The minimum distance* $d = 2^{k-1}$*, where k is the dimension the code.*
- *The code length* $n = 2^k - 1 = 2d - 1$.
- *All of its nonzero codewords have Hamming weights equal to d.*

Proposition 3. *By recursively puncturing a Reed-Muller (RM) code's generator matrix using the proposed algorithm based on the Ashikhmin-Barg inequality, the resulting MICODE is equivalent to a binary simplex code.*

Proposition 4. *Any Reed-Muller code of minimum distance d generates a MICODE of length* $n' = 2d-1$ *and dimension* $k' = \log_2(d) + 1$.

Proof. Given a Reed-Muller code of minimum distance d, the minimal code generated by our algorithm is equivalent to a simplex code. Since the minimum distance is preserved in the generated minimal code, and the length and dimension of a simplex code are determined by its minimum distance d, it follows that any RM code with minimum distance d, irrespective of its original length and dimension, produces a minimal code with parameters $n' = 2d-1$ and $k' = \log_2(d) + 1$ under our construction. □

Table 2 displays the minimal code parameters derived from a set of Reed-Muller codes, as well as the number of iterations I required to move from C to C'. The importance of the minimum distance of a code

Table 2: The generated minimal code from the original Reed-Muller code.

The original Reed-Muller code				The generated minimal code			I	I_{max}
$RM(m,r)$	n	k	d	n'	k'	d		
$RM(3,1)$	8	4	4	7	3	4	1	1
$RM(3,2)$	8	7	2	3	2	2	5	5
$RM(4,1)$	16	5	8	15	4	8	1	1
$RM(4,2)$	16	11	4	7	3	4	8	9
$RM(5,1)$	32	6	16	31	5	16	1	1
$RM(5,2)$	32	16	8	15	4	8	12	14
$RM(5,3)$	32	26	4	7	3	4	23	24
$RM(5,4)$	32	31	2	3	2	2	29	29
$RM(6,1)$	64	7	32	63	6	32	1	1
$RM(6,2)$	64	22	16	31	5	16	17	20
$RM(6,3)$	64	42	8	15	4	8	38	40
$RM(7,1)$	128	8	64	127	7	64	1	1
$RM(7,2)$	128	29	32	63	6	32	23	27
$RM(7,3)$	128	64	16	31	5	16	59	62
$RM(7,4)$	128	99	8	15	4	8	95	97
$RM(8,2)$	256	37	64	127	7	64	30	35
$RM(8,3)$	256	93	32	63	6	32	87	91
$RM(8,6)$	256	247	4	7	3	4	244	245
$RM(10,2)$	1024	56	256	511	9	256	47	54
$RM(10,3)$	1024	176	128	255	8	128	168	174
$RM(11,2)$	2048	67	512	1023	10	512	57	65

is widely recognized in the context of error correction and detection. A large minimum distance enhances the error-correcting capability of a code. However, the minimal codes derived in our work tend to have a large minimum distance at the expense of a reduced code rate. This trade-off makes such minimal codes less practical for real-time communication systems, where achieving a balance between error performance and code rate is crucial. While our MICODE may have limited practicality for real-time communication due to its poor code rate, offers a large minimum distance, making it ideal for cryptographic applications, such as Massey's secret sharing (Massey, 1993; Massey, 1995). We will show how MICODE can enhance the security of such schemes.

4 PROPOSED MASSEY SCHEME BASED ON MICODE

This section describes Massey's secret sharing scheme (Massey, 1993; Massey, 1995). We define the minimal access structure of the secret sharing based on the dual of the generated MICODE and the number of authorized minimal access sets.

4.1 Massey's Secret Sharing

Consider $\mathbf{P} = \{p_1, ..., p_{n-1}\}$ to be a set of $n-1$ participants, $\Gamma \subseteq 2^{\mathbf{P}}$ to be the collection of all access sets known as the access structure and $\Gamma^m \subset \Gamma$ to be the collection of all minimal access sets known as the minimal access structure. The Massey's secret sharing works as follows: Let C be an $[n,k]_q$ linear code over $\mathbb{F}_q$ and $G = [g_0, ..., g_{n-1}]$ its generator matrix. Let $s \in \mathbb{F}_q$ be the secret and $m \in \mathbb{F}_q^k$ be the message vector that satisfies $s = mg_0$. In the secret sharing based on C the dealer computes $\mathbf{s} = mG = (s, s_1, ..., s_{n-1})$, and then each participant p_i receives the share s_i, $i = \{1, ..., n-1\}$.

A group of participants $\{p_{i_1}, p_{i_2}, ..., p_{i_t}\}$ can recover s if and only if g_0 is a linear combination of $\{g_{i_1}, g_{i_2}, ..., g_{i_t}\}$, $0 < i_1 < ... < i_t \leq n-1$, i.e. there exists a codeword $c' = (1, 0, .., c'_{i_1}, ..., 0, ..., c'_{i_t}, ..., 0) \in C^{\perp}$ provided $c'_{i_j} \neq 0$ for at least one j. If there exists such a codeword, then the secret s can be computed as follows (Ding and Yuan, 2003):

- Determine $x = (x_1, ..., x_t)$ by solving

$$\sum_{j=1}^{t} x_j g_{i_j} = g_0. \tag{11}$$

- Then compute

$$s = \sum_{j=1}^{t} x_j s_{i_j}. \quad (12)$$

Theorem 2. *Let C be a binary $RM(p,r)$ code and C' the $[n',k',d]$ MICODE generated from C. There are $2^{k'-1}$ minimal access sets in the SSS based on $C'^{\perp}$.*

Proof. Assume G' is the generator matrix of the generated binary $[n',k',d]$ minimal code, and each row of G' has one in the first coordinate; there are $2^{k'}$ minimal codewords. According to (Massey, 1993) each minimal codeword whose first coordinate equal to one is a minimal access set; thus, each information vector m with an odd number of non-zero coordinates produces a minimal codeword with the first coordinate equal to one. Thus, there are $2^{k'-1}$ minimal codewords with the first coordinate equal to one from all possible $2^{k'}$ information vectors, implying $2^{k'-1}$ minimal access sets. □

Definition 9 (Dictatorial participant)**.** *Let $\boldsymbol{P} = \{p_1,...,p_{n-1}\}$ be a set of $n-1$ participants. A participant $p_i, 0 < i < n$ is called dictatorial if it serves in every minimal access set (Ding and Yuan, 2003).*

Theorem 3. *Let $G = [g_0,...,g_{n-1}]$ be a generator matrix of a $[n,k]_q$ minimal code, and $\boldsymbol{P} = \{p_1,...,p_{n-1}\}$ is the set of participants. If $g_i, 0 < i < n$ is a multiple of g_0, then the participant p_i is dictatorial. Otherwise, if g_i is not a multiple of g_0, the participant must be in $(q-1)q^{k-2}$ out of the q^{k-1} possible minimal access sets (Ding and Yuan, 2003).*

Remark 1. *In our proposal, the target code is a binary Reed-Muller C, and the generated minimal code C' is also binary, which means that all the columns of C' are distinct and no column is a multiple of the other, implying that no dictatorial participant exists. It is worth noting that each participant $p_i \in \boldsymbol{P}, 0 < i < n$ appears in 2^{k-2} out of the 2^{k-1} possible minimal access sets.*

Example 4. *Suppose the generator matrix G_8 of the $[7,3,4]$-minimal code C'. In the SSS based on $C'^{\perp}$, from the set of shares $\{s_1,...,s_6\}$, the 4 possible minimal access sets are: $\{1,2,3\}$, $\{1,4,5\}$, $\{2,4,6\}$,$\{3,5,6\}$*

Example 5. *Table 3 shows the minimal access sets corresponding to some of the generated MICODEs.*

5 ANALYSIS AND EVALUATION

The security of Massey secret sharing based on a code C depends significantly on the minimum distance of $C^{\perp}$. A. Renvall and C. Ding have demonstrated the following theorem in their publisher paper (Renvall and Ding, 1996).

Theorem 4. *If G is a generator matrix of a $[n,k]_q$ linear code over $\mathbb{F}_q$, then any coalition of $d^{\perp}-2$ or fewer participants cannot determine the secret in the secret sharing scheme based on C for Massey construction, and there is at least a group of $d^{\perp}-1$ participants that can recover the secret, where $d^{\perp}$ is the minimum distance of the dual code of C.*

Thus, the access structure and security of the Massey secret sharing scheme are entirely dependent on the structure of the dual code $C^{\perp}$. As a result, the access structure in our proposal is defined by the generated $[n',k',d]$-MICODE. For meaningful security, it should consider the minimum distance d of the generated minimal code. An attacker should be unable to reconstruct the secret if they possess fewer than $d-1$ shares. Nevertheless, there is a scenario where the attacker can find a specific set of $d-1$ shares that reveals the secret. By finding a solution $x = (x_1,...,x_t)$ that satisfies the equation 11 where $w_H(x) = d-1$, the attacker can successfully determine the secret through 2^{d-1} possible guesses. As a result, a code with a high minimum distance is preferable for secret sharing because it ensures that the secret is secure even if some of the shares are compromised. This assertion strengthens the superiority of our proposed minimal code over certain existing works, like the one presented in (Ding et al., 2018). For example, in (Ding et al., 2018), the authors introduced a minimal code with length 63 and minimum distance 14. In contrast, our proposed minimal code achieves a minimum distance of 32 with the same code length. Table 4 provides a comparative analysis between our proposed minimal code and the one presented in (Ding et al., 2018) for various code lengths.

Proposition 5. *In the SSS based on the dual of our MICODE, all the minimal access sets have the same size.*

Proof. All nonzero codewords of a simplex code have the same Hamming weight. Since our MICODE is equivalent to a simplex code, it follows that in the secret sharing scheme based on the dual of our MICODE, all minimal access sets have the same size. □

5.1 Comparison of Our Secret Sharing Scheme with Shamir's Scheme

This section compares our secret sharing scheme with Shamir's Secret Sharing (Shamir, 1979), emphasizing key advantages. Although Shamir's scheme is

Table 3: The minimal access sets corresponding to some of the generated MICODEs.

MICODE $[n',k',d]$	Minimal Access Sets
$[3,2,2]$	$\{2\}, \{1\}$
$[7,3,4]$	$\{2,4,6\}, \{1,4,5\}, \{1,2,3\}, \{3,5,6\}$
$[15,4,8]$	$\{2,4,6,8,10,12,14\}$, $\{1,4,5,8,9,12,13\}$, $\{1,2,3,8,9,10,11\}$, $\{3,5,6,8,11,13,14\}$, $\{1,2,3,4,5,6,7\}$, $\{3,4,7,9,10,13,14\}$, $\{2,5,7,9,11,12,14\}$, $\{1,6,7,10,11,12,13\}$
$[31,5,16]$	$\{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30\}$, $\{1,4,5,8,9,12,13,16,17,20,21,24,25,28,29\}$, $\{1,2,3,8,9,10,11,16,17,18,19,24,25,26,27\}$, $\{3,5,6,8,11,13,14,16,19,21,22,24,27,29,30\}$, $\{1,2,3,4,5,6,7,16,17,18,19,20,21,22,23\}$, $\{3,4,7,9,10,13,14,16,19,20,23,25,26,29,30\}$, $\{2,5,7,9,11,12,14,16,18,21,23,25,27,28,30\}$, $\{1,6,7,10,11,12,13,16,17,22,23,26,27,28,29\}$, $\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$, $\{3,4,7,8,11,12,15,17,18,21,22,25,26,29,30\}$, $\{2,5,7,8,10,13,15,17,19,20,22,25,27,28,30\}$, $\{1,6,7,8,9,14,15,18,19,20,21,26,27,28,29\}$, $\{2,4,6,9,11,13,15,17,19,21,23,24,26,28,30\}$, $\{1,4,5,10,11,14,15,18,19,22,23,24,25,28,29\}$, $\{1,2,3,12,13,14,15,20,21,22,23,24,25,26,27\}$, $\{3,5,6,9,10,12,15,17,18,20,23,24,27,29,30\}$
$[63,6,32]$	$\{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62\}$, $\{1,4,5,8,9,12,13,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,48,49,52,53,56,57,60,61\}$, $\{1,2,3,8,9,10,11,16,17,18,19,24,25,26,27,32,33,34,35,40,41,42,43,48,49,50,51,56,57,58,59\}$, $\{3,5,6,8,11,13,14,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,48,51,53,54,56,59,61,62\}$, $\{1,2,3,4,5,6,7,16,17,18,19,20,21,22,23,32,33,34,35,36,37,38,39,48,49,50,51,52,53,54,55\}$, $\{3,4,7,9,10,13,14,16,19,20,23,25,26,29,30,32,35,36,39,41,42,45,46,48,51,52,55,57,58,61,62\}$, $\{2,5,7,9,11,12,14,16,18,21,23,25,27,28,30,32,34,37,39,41,43,44,46,48,50,53,55,57,59,60,62\}$, $\{1,6,7,10,11,12,13,16,17,22,23,26,27,28,29,32,33,38,39,42,43,44,45,48,49,54,55,58,59,60,61\}$, $\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47\}$, $\{3,4,7,8,11,12,15,17,18,21,22,25,26,29,30,32,35,36,39,40,43,44,47,49,50,53,54,57,58,61,62\}$, $\{2,5,7,8,10,13,15,17,19,20,22,25,27,28,30,32,34,37,39,40,42,45,47,49,51,52,54,57,59,60,62\}$, $\{1,6,7,8,9,14,15,18,19,20,21,26,27,28,29,32,33,38,39,40,41,46,47,50,51,52,53,58,59,60,61\}$, $\{2,4,6,9,11,13,15,17,19,21,23,24,26,28,30,32,34,36,38,41,43,45,47,49,51,53,55,56,58,60,62\}$, $\{1,4,5,10,11,14,15,18,19,22,23,24,25,28,29,32,33,36,37,42,43,46,47,50,51,54,55,56,57,60,61\}$, $\{1,2,3,12,13,14,15,20,21,22,23,24,25,26,27,32,33,34,35,44,45,46,47,52,53,54,55,56,57,58,59\}$, $\{3,5,6,9,10,12,15,17,18,20,23,24,27,29,30,32,35,37,38,41,42,44,47,49,50,52,55,56,59,61,62\}$, $\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31\}$, $\{3,4,7,8,11,12,15,16,19,20,23,24,27,28,31,33,34,37,38,41,42,45,46,49,50,53,54,57,58,61,62\}$, $\{2,5,7,8,10,13,15,16,18,21,23,24,26,29,31,33,35,36,38,41,43,44,46,49,51,52,54,57,59,60,62\}$, $\{1,6,7,8,9,14,15,16,17,22,23,24,25,30,31,34,35,36,37,42,43,44,45,50,51,52,53,58,59,60,61\}$, $\{2,4,6,9,11,13,15,16,18,20,22,25,27,29,31,33,35,37,39,40,42,44,46,49,51,53,55,56,58,60,62\}$, $\{1,4,5,10,11,14,15,16,17,20,21,26,27,30,31,34,35,38,39,40,41,44,45,50,51,54,55,56,57,60,61\}$, $\{1,2,3,12,13,14,15,16,17,18,19,28,29,30,31,36,37,38,39,40,41,42,43,52,53,54,55,56,57,58,59\}$, $\{3,5,6,9,10,12,15,16,19,21,22,25,26,28,31,33,34,36,39,40,43,45,46,49,50,52,55,56,59,61,62\}$, $\{2,4,6,8,10,12,14,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,48,50,52,54,56,58,60,62\}$, $\{1,4,5,8,9,12,13,18,19,22,23,26,27,30,31,34,35,38,39,42,43,46,47,48,49,52,53,56,57,60,61\}$, $\{1,2,3,8,9,10,11,20,21,22,23,28,29,30,31,36,37,38,39,44,45,46,47,48,49,50,51,56,57,58,59\}$, $\{3,5,6,8,11,13,14,17,18,20,23,25,26,28,31,33,34,36,39,41,42,44,47,48,51,53,54,56,59,61,62\}$, $\{1,2,3,4,5,6,7,24,25,26,27,28,29,30,31,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55\}$, $\{3,4,7,9,10,13,14,17,18,21,22,24,27,28,31,33,34,37,38,40,43,44,47,48,51,52,55,57,58,61,62\}$, $\{2,5,7,9,11,12,14,17,19,20,22,24,26,29,31,33,35,36,38,40,42,45,47,48,50,53,55,57,59,60,62\}$, $\{1,6,7,10,11,12,13,18,19,20,21,24,25,30,31,34,35,36,37,40,41,46,47,48,49,54,55,58,59,60,61\}$

Table 4: Our MICODE vs. the minimal code presented in (Ding et al., 2018).

The minimal code (Ding et al., 2018)			Our MICODE		
n	k	d	n	k	d
63	7	14	63	6	32
127	8	52	127	7	64
255	9	60	255	8	128
511	10	224	511	9	256
1023	11	248	1023	10	512

commonly used for threshold-based secret sharing, it lacks flexibility in access structure design and does not inherently provide enhanced security for applications requiring strict security and customizable access policies. Furthermore, Shamir's scheme requires that the field $\mathbb{F}_q$ be larger than the number of shares n. Consequently, each share must be represented using at least $\log_2(n)$ bits, resulting in inefficiencies in both communication and storage. Table 5 provides a detailed comparison, highlighting key differences between the two schemes.

Table 5: Our Secret Sharing vs. Shamir's Secret Sharing (Shamir, 1979).

Scheme	Our scheme	Shamir's scheme
Mathematical Basis	Linear algebra and coding theory	Polynomial interpolation over finite field $\mathbb{F}_q$
Access Structure	Flexible (based on $[n,k,d]$ MICODE)	Fixed threshold k (any k or more shares)
Share Size (Bits)	1	$O(\log n)$
Secret Reconstruction	Requires exactly $d-1$ out of $n-1$ shares (only those in the minimal access set)	Requires at least k out of n shares
Resilience to Share Compromise	More resilient (secret remains secure unless specific shares are compromised)	Less resilient (secret can be reconstructed if any k shares are compromised)

6 CONCLUSIONS

The main goals of this paper are to generate a minimal code from a Reed-Muller code with good and flexible parameters and then to derive a secret sharing scheme from the generated minimal code. There are some existing works to determine the minimal codewords, i.e. (solving the covering problem), in a specific class of codes, such as the Reed-Muller code. However, in the case of Reed-Muller code, this has only been solved for the second-order binary Reed-Muller code, and the problem for general code, including Reed-Muller code, remains difficult. In contrast, we have used the Ashikhmin-Barg inequality to efficiently construct a minimal code (called MICODE) from the Reed-Muller code while preserving the minimum distance. While our MICODE exhibits a reduced code rate, it offers substantial advantages for cryptographic applications, particularly in secret sharing schemes like Massey's secret sharing, where the emphasis is on enhancing security through a large minimum distance, rather than optimizing for high-rate communication.

This work opens up new possibilities for enhancing the security of SSS protocols by using codes with large minimum distances. Future work could focus on further improving the construction techniques to optimize the code rate while maintaining a high minimum distance. Additionally, exploring the application of these minimal codes in other cryptographic contexts, such as secure multiparty computation, would offer valuable insights into their broader applicability.

ACKNOWLEDGEMENT

This work was supported by the European Commission under the Horizon Europe Programme, as part of the project LAZARUS (https://lazarus-he.eu/) (Grant Agreement no. 101070303). The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

REFERENCES

Abbe, E., Shpilka, A., and Ye, M. (2020). Reed–muller codes: Theory and algorithms. *IEEE Transactions on Information Theory*, 67(6):3251–3277.

Agrell, E. (1996). Voronoi regions for binary linear block codes. *IEEE Transactions on Information Theory*, 42(1):310–316.

Agrell, E. (1998). On the voronoi neighbor ratio for binary linear block codes. *IEEE Transactions on Information Theory*, 44(7):3064–3072.

Ashikhmin, A. and Barg, A. (1998). Minimal vectors in linear codes. *IEEE Transactions on Information Theory*, 44(5):2010–2017.

Ball, S. and Blokhuis, A. (2013). A bound for the maximum weight of a linear code. *SIAM Journal on Discrete Mathematics*, 27(1):575–583.

Blakley, G. R. (1979). Safeguarding cryptographic keys. In *Managing requirements knowledge, international workshop on*, pages 313–313. IEEE Computer Society.

Chang, S. and Hyun, J. Y. (2018). Linear codes from simplicial complexes. *Designs, Codes and Cryptography*, 86:2167–2181.

Ding, C., Heng, Z., and Zhou, Z. (2018). Minimal binary linear codes. *IEEE Transactions on Information Theory*, 64(10):6536–6545.

Ding, C., Kohel, D. R., and Ling, S. (2000). Secret-sharing with a class of ternary codes. *Theoretical Computer Science*, 246(1-2):285–298.

Ding, C. and Salomaa, A. (2006). Secret sharing schemes with nice access structures. *Fundamenta Informaticae*, 73(1-2):51–63.

Ding, C. and Yuan, J. (2003). Covering and secret sharing with linear codes. In *International Conference on Discrete Mathematics and Theoretical Computer Science*, pages 11–25. Springer.

Helleseth, T., Klove, T., and Levenshtein, V. I. (2004). The simplex codes and other even-weight binary linear codes for error correction. *IEEE transactions on information theory*, 50(11):2818–2823.

Li, Z., Xue, T., and Lai, H. (2010). Secret sharing schemes from binary linear codes. *Information Sciences*, 180(22):4412–4419.

Massey, J. L. (1993). Minimal codewords and secret sharing. In *Proceedings of the 6th joint Swedish-Russian international workshop on information theory*, pages 276–279.

Massey, J. L. (1995). Some applications of coding theory in cryptography. *Codes and Ciphers: Cryptography and Coding IV*, pages 33–47.

McEliece, R. J. and Sarwate, D. V. (1981). On sharing secrets and reed-solomon codes. *Communications of the ACM*, 24(9):583–584.

Renvall, A. and Ding, C. (1996). The access structure of some secret-sharing schemes. In *Information Security and Privacy: First Australasian Conference, ACISP'96 Wollongong, NSW, Australia, June 24–26, 1996 Proceedings 1*, pages 67–78. Springer.

Schillewaert, J., Storme, L., and Thas, J. A. (2010). Minimal codewords in reed–muller codes. *Designs, Codes and Cryptography*, 54:273–286.

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

Song, Y. and Li, Z. (2012). Secret sharing with a class of minimal linear codes. *arXiv preprint arXiv:1202.4058*.

Xu, G. and Qu, L. (2019). Three classes of minimal linear codes over the finite fields of odd characteristic. *IEEE Transactions on Information Theory*, 65(11):7067–7078.

Yuan, J. and Ding, C. (2005). Secret sharing schemes from three classes of linear codes. *IEEE transactions on information theory*, 52(1):206–212.