Notes on BaseFold (Part IV): Random Foldable Codes

- Jade Xie jade@secbit.io
- Yu Guo yu.guo@secbit.io

Previous articles have mentioned that BaseFold extends the FRI IOPP by introducing the concept of *foldable codes*. Additionally, by combining the Sumcheck protocol, it can support PCS for multi-linear polynomials. The next crucial question is how to explicitly construct such *foldable codes*. We aim for these foldable codes to possess the following properties:

1. **Efficient Encoding**
2. **Field Agnostic**, i.e., applicable even for small fields
3. **Compatible with PCS for Multi-linear Polynomials**

Another important aspect of encoding is the consideration of the Minimum Relative Hamming Distance. If readers are familiar with the FRI protocol, the Reed-Solomon codes used are likely not unfamiliar. They have a desirable property: their distance meets the Singleton bound, i.e., $d = n - k + 1$, and are thus known as Maximum Distance Separable (MDS) codes. These codes balance code length and error-correcting capability effectively, providing strong error detection and correction with minimal redundancy, thereby saving encoding space. In PCS protocols, this allows verifiers to perform checks more efficiently. Therefore, from a practical perspective, we also desire that such foldable codes satisfy the fourth property:

4. **Good Relative Minimum Distance**

The BaseFold paper [ZCF23] constructs a type of code called *Random Foldable Code* (RFCs), which satisfies the aforementioned properties. Next, we will explore how it achieves these points.

Efficient Encoding Algorithms

The first article in this series has already introduced the concept of *foldable linear codes* and the BaseFold encoding algorithm. Here is a brief review.

Definition 1 [ZCF23, Definition 5] ((c, k_0, d) - Foldable Linear Codes). Let $c, k_0, d \in \mathbb{N}$ and $\mathbb{F}$ denote a finite field. A linear code $C_d : \mathbb{F}^{k_0 \cdot 2^d} \to \mathbb{F}^{ck_0 \cdot 2^d}$ with generator matrix $\mathbf{G}_d$ is called ***foldable*** if there exists a sequence of generator matrices $(\mathbf{G}_0, \ldots, \mathbf{G}_{d-1})$ and diagonal matrices $(T_0, \ldots, T_{d-1})$ and $(T'_0, \ldots, T'_{d-1})$ such that for any $i \in [1, d]$, the following holds:

1. The diagonal matrices $T_{i-1}, T'_{i-1} \in F^{ck_0 \cdot 2^{i-1} \times ck_0 \cdot 2^{i-1}}$ satisfy $\mathrm{diag}(T_{i-1})[j] \neq \mathrm{diag}(T'_{i-1})[j]$ for all $j \in [ck_0 \cdot 2^{i-1}]$;
2. The matrix $\mathbf{G}_i \in F^{k_0 \cdot 2^i \times ck_0 \cdot 2^i}$ (arranged row-wise) is equal to

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{G}_{i-1} & \mathbf{G}_{i-1} \\ \mathbf{G}_{i-1} \cdot T_{i-1} & \mathbf{G}_{i-1} \cdot T'_{i-1} \end{bmatrix}. \tag{1}$$

To efficiently construct a foldable linear code, a uniform sampling method is employed by first defining a set of random foldable distributions.

Definition 2 [ZCF23, Definition 9] ((c, k_0) - Foldable Distributions). Fix a finite field $\mathbb{F}$ and $c, k_0 \in \mathbb{N}$. Let $\mathbf{G}_0 \in \mathbb{F}^{k_0 \times ck_0}$ be the generator matrix of an $[ck_0, k_0]$ linear code that satisfies maximum distance separability, and let D_0 be the distribution that outputs $\mathbf{G}_0$ with probability 1. For each $i > 0$, we recursively define the distribution D_i, which samples the generator matrices $(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_i)$ where $\mathbf{G}_i \in F^{k_i \times n_i}$ with $k_i := k_0 \cdot 2^i$, $n_i := ck_i$:

1. Sample $(\mathbf{G}_0, \ldots, \mathbf{G}_{i-1}) \leftarrow D_{i-1}$;
2. Sample $\mathrm{diag}(T_{i-1}) \leftarrow \$(\mathbb{F}^\times)^{n_{i-1}}$ and define $\mathbf{G}_i$ as

$$\mathbf{G}_i = \begin{bmatrix} \mathbf{G}_{i-1} & \mathbf{G}_{i-1} \\ \mathbf{G}_{i-1} \cdot T_{i-1} & \mathbf{G}_{i-1} \cdot -T_{i-1} \end{bmatrix}. \tag{2}$$

Once the initial generator matrix $\mathbf{G}_0$ is determined, uniformly sample n_0 random elements from $\mathbb{F}^\times$ (i.e., excluding the zero element) to generate the diagonal elements of T_0, thereby obtaining the next generator matrix $\mathbf{G}_1$. This process is then recursively applied to generate $(\mathbf{G}_2, \ldots, \mathbf{G}_i)$. In PCS, generating foldable codes via uniform sampling aids in achieving an efficient prover.

Note that the above definition requires the initial $\mathbf{G}_0$ to be the generator matrix of a linear code satisfying the MDS property. However, as mentioned in a footnote in [ZCF23], this requirement is not strictly necessary. Including this property is merely for simplifying the analysis of the code distance later on. In fact, the distance analysis holds for any linear code.

Protocol 1 Enc_d [ZCF23, Protocol 1]: BaseFold Encoding Algorithm

Input: Original message $\mathbf{m} \in \mathbb{F}^{k_d}$

Output: $\mathbf{w} \in \mathbb{F}^{n_d}$ such that $\mathbf{w} = \mathbf{m} \cdot \mathbf{G}_d$

Parameters: $\mathbf{G}_0$ and diagonal matrices $(T_0, T_1, \ldots, T_{d-1})$

1. If $d = 0$ (i.e., $\mathbf{m} \in \mathbb{F}^{k_0}$):
   - (a) Return $\mathrm{Enc}_0(\mathbf{m})$
2. Else:
   - (a) Split $\mathbf{m} := (\mathbf{m}_l, \mathbf{m}_r)$
   - (b) Let $\mathbf{l} := \mathrm{Enc}_{d-1}(\mathbf{m}_l)$, $\mathbf{r} := \mathrm{Enc}_{d-1}(\mathbf{m}_r)$, and $\mathbf{t} = \mathrm{diag}(T_{d-1})$
   - (c) Return $(\mathbf{l} + \mathbf{t} \circ \mathbf{r}, \mathbf{l} - \mathbf{t} \circ \mathbf{r})$

By analyzing Protocol 1, we can see that encoding to obtain C_d requires only $\frac{dn_d}{2}$ field multiplications and dn_d field additions, i.e., $0.5n \log n$ field multiplications and $n \log n$ field additions. Overall, the encoding complexity is $O(n \log n)$. Thus, we have introduced the explicit construction of Random Linear Foldable Codes provided by BaseFold and verified that they indeed support efficient encoding.

Polynomials Save the World: Polynomial-Based Encoding

Next, we examine the second and third properties of *Random Foldable Codes*:

2. **Field Agnostic**, i.e., applicable even for small fields
3. **Compatible with PCS for Multi-linear Polynomials**

As mentioned earlier, Reed-Solomon codes can achieve the Singleton bound, but only when the alphabet size is relatively large (i.e., $q \gg n$). Fortunately, we can extend Reed-Solomon codes to Reed-Muller codes, transitioning from univariate polynomial codes to multivariate polynomial codes. This allows applicability over small fields ($q \ll n$), although there is a slight trade-off in the balance between distance and error-correcting capability. However, this is worthwhile.

Appendix D of [ZCF23] informs us that *Random Foldable Codes* are a special case of truncated Reed-Muller codes (Punctured Reed-Muller Codes). Thus, *Random Foldable Codes* are field agnostic and, by venturing into the realm of multivariate polynomials, are suitable for PCS of multi-linear polynomials.

We know that the encoding space of Reed-Solomon codes consists of univariate polynomials of degree at most d. Reed-Muller codes extend this to multivariate polynomials, with the encoding space comprising multivariate polynomials of total degree at most d.

For n, d, q with $d < q$, define Reed-Muller encoding as ([GJX15])

$$\mathrm{RM}_q(n, d) := \{(F(\mathbf{u}))_{\mathbf{u} \in \mathbb{F}_q^n} : F \in \mathbb{F}_q[X_1, \cdots, X_n], \deg(F) \leq d\}. \tag{3}$$

Reed-Muller codes represent the set of evaluations of n-variate polynomials of total degree at most d over $\mathbb{F}_q^n$. The length of the encoding $\mathrm{RM}_q(n,d)$ is q^n, and the dimension is $\binom{n+d}{n}$.

Intuitively, Punctured Reed-Muller Codes are simply Reed-Muller codes with truncation. Specifically, the evaluation points $\mathbf{u}$ are not taken from all of $\mathbb{F}_q^n$ but only a subset, denoted as $\mathcal{T} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$. Typically, Punctured Reed-Muller Codes allow this set $\mathcal{T}$ to be a multiset (i.e., permitting duplicate elements), but we do not impose this requirement here. Let $\mathrm{RM}_q(n,d)|_{\mathcal{T}}$ denote the $\mathbb{F}_q$-linear code:

$$\mathrm{RM}_q(n,d)|_{\mathcal{T}} := \{(F(\mathbf{u}_1), F(\mathbf{u}_2), \cdots, F(\mathbf{u}_N)) : F \in \mathbb{F}_q[X_1, \cdots, X_n], \deg(F) \leq d\}. \tag{4}$$

This is called a punctured Reed-Muller code ([GJX15]). From the definition, it is evident that this is merely a subset of Reed-Muller codes, selecting only N points, which aligns with the literal meaning of "truncated" or "punctured."

With the concept of Punctured Reed-Muller Codes established, let's examine the following lemma provided in Appendix D of [ZCF23], which states that foldable linear codes are a special case of punctured Reed-Muller codes.

Lemma 1 [ZCF23, Lemma 11] (Foldable Punctured Reed-Muller Codes). Let C_d be a foldable linear code with generator matrices $(\mathbf{G}_0, \ldots, \mathbf{G}_{d-1})$ and diagonal matrices $(T_0, \ldots, T_{d-1})$, $(T'_0, \ldots, T'_{d-1})$. Then there exists a subset $D \subset \mathbb{F}^d$ such that $C_d = \{(P(\mathbf{x}) : \mathbf{x} \in D) : P \in \mathbb{F}[X_1, \ldots, X_d]\}$, i.e., each codeword in C_d is a vector obtained by evaluating a multilinear polynomial P at each point in D.

Proof: By induction. For simplicity, consider C_0 as a repetition code. In the base case, $\mathrm{Enc}_0(m) = m\| \ldots \|m$ is a constant polynomial $P \equiv m$ evaluated at c distinct points. Assume that for $i < d$, there exists a set D_i such that $C_i = \{(P(\mathbf{x}) : \mathbf{x} \in D_i) : P \in \mathbb{F}[X_1, \ldots, X_i]\}$. Without loss of generality, assign an integer $j \in [1, c \cdot 2^i]$ to index each element in D_i sequentially, representing x_j as the j-th element in D_i.

Let $t = \mathrm{diag}(T_i)$, $t' = \mathrm{diag}(T'_i)$, $n_i = c \cdot 2^i$, $\mathbf{v} \in \mathbb{F}^{2^{i+1}}$, and let $P \in \mathbb{F}[X_1, \ldots, X_{i+1}]$ be a polynomial with coefficients from $\mathbf{v}$. Finally, let $P_l, P_r \in \mathbb{F}[X_1, \ldots, X_i]$ such that $P(X_1, \ldots, X_{i+1}) = P_l(X_1, \ldots, X_i) + X_{i+1}P_r(X_1, \ldots, X_i)$. Then,

$$\begin{aligned}
\mathrm{Enc}_{i+1}(\mathbf{v}) &= \mathrm{Enc}_i(\mathbf{v}_l) + \mathrm{diag}(T_i) \circ \mathrm{Enc}_i(\mathbf{v}_r) \quad \| \quad \mathrm{Enc}_i(\mathbf{v}_l) + \mathrm{diag}(T_i) \circ \mathrm{Enc}_i(\mathbf{v}_r) \\
&\quad \text{(by the encoding algorithm in Protocol 1)} \\
&= (P_l(\mathbf{x}_1), \ldots, P_l(\mathbf{x}_n)) + \mathrm{diag}(T_i) \circ (P_r(\mathbf{x}_1), \ldots, P_r(\mathbf{x}_n)) \\
&\| \quad (P_l(\mathbf{x}_1), \ldots, P_l(\mathbf{x}_n)) + \mathrm{diag}(T'_i) \circ (P_r(\mathbf{x}_1), \ldots, P_r(\mathbf{x}_n)) \\
&\quad \text{(by the induction hypothesis)} \\
&= (P_l(\mathbf{x}_1) + t_1 P_r(\mathbf{x}_1), \ldots, P_l(\mathbf{x}_n) + t_n P_r(\mathbf{x}_n), P_l(\mathbf{x}_1) + t'_1 P_r(\mathbf{x}_1), \ldots, P_l(\mathbf{x}_n) + t'_n P_r(\mathbf{x}_n)) \\
&\quad \text{(by the definition of the Hadmard product)} \\
&= (P(\mathbf{x}_1, t_1), \ldots, P(\mathbf{x}_n, t_n), P(\mathbf{x}_1, t'_1), \ldots, P(\mathbf{x}_n, t'_n)) \\
&\quad \text{(by the definition of } P\text{)}
\end{aligned} \tag{5}$$

Therefore, let $D_{i+1} = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_n, t_n), (\mathbf{x}_1, t'_1), \ldots, (\mathbf{x}_n, t'_n)\}$, and the lemma holds for $i+1$. Thus, by induction, the proof is complete. □

Good Relative Minimum Distance

Finally, we focus on the fourth property satisfied by *Random Foldable Codes* (RFCs):

4. **Good Relative Minimum Distance**

In [ZCF23], it is proven that RFCs have tight bounds on their minimum Hamming distance (a "tight" bound means that the actual bounds are achievable). For example, an RFC over a 256-element finite field with a message length of 2^{25} and a code rate of $\frac{1}{8}$ has a relative minimum distance of 0.728 with overwhelming probability. For a code with a rate of $\frac{1}{8}$, the maximum achievable relative minimum Hamming distance is approximately $1 - \frac{1}{8} = 0.875$. Clearly, 0.728 is fairly close to 0.875. This is practically useful and implies that foldable codes generated via overwhelming probability (which enable efficient PCS provers) also have good relative minimum distances (enabling efficient PCS verifiers).

The term "overwhelming probability" arises from the distribution $(\mathbf{G}_0, \ldots, \mathbf{G}_d) \leftarrow D_d$ introduced during encoding. When uniformly sampling diagonal matrices T_i from $\mathbb{F}^\times$ and setting $T_i' = -T_i$, the relative minimum distance of C_d achieved with overwhelming probability is equal to

$$1 - \left(\frac{\epsilon_{\mathbb{F}}^d}{c} + \frac{\epsilon_{\mathbb{F}}}{\log |\mathbb{F}|} \sum_{i=0}^{d} (\epsilon_{\mathbb{F}})^{d-i} \left(0.6 + \frac{2\log(n_i/2) + \lambda}{n_i} \right) \right) \tag{1}$$

where c is the reciprocal of the code rate, $\epsilon_{\mathbb{F}} = \frac{\log |\mathbb{F}|}{\log |\mathbb{F}| - 1.001}$, n_i is the encoding length, d is the logarithm of the message length, and λ is the security parameter. By setting $\lambda = 128$, it ensures that (c, k_0, d)-random foldable linear codes achieve the above relative minimum distance with a probability of at least $1 - 2^{-128}$.

Next, we examine how the result in equation (1) is derived. Our goal is to analyze the relative minimum distance of foldable random codes C_d. For a linear code, the minimum distance equals the minimum Hamming weight among all non-zero codewords because

$$d = \min_{\substack{\vec{c}_1 \neq \vec{c}_2 \\ \vec{c}_1, \vec{c}_2 \in C_d}} \Delta(\vec{c}_1, \vec{c}_2) = \min_{\substack{\vec{c}_1 \neq \vec{c}_2 \\ \vec{c}_1, \vec{c}_2 \in C_d}} wt(\vec{c}_1 - \vec{c}_2) = \min_{\vec{c} \neq \vec{0}, \vec{c} \in C_d} wt(\vec{c}) \tag{6}$$

Since it is a linear code, $\vec{c}_1 - \vec{c}_2$ is also a codeword in C_d, hence the last equality holds. Therefore, we want to show that for any non-zero message, i.e., $\forall \vec{m} \neq \vec{0}$, the encoded codeword $\mathrm{Enc}_d(\vec{m})$ does not have too many zero components. Suppose it has at most t_d zero components, letting $\mathrm{nzero}(\cdot)$ denote the number of zero components in a vector, we aim to show

$$\forall \vec{m} \neq \vec{0}, \quad \mathrm{nzero}(\mathrm{Enc}_d(\vec{m})) \leq t_d \tag{2}$$

Let n_d denote the length of the codeword $\mathrm{Enc}_d(\vec{m})$. Then from (2), we have

$$\forall \vec{m} \neq \vec{0}, \quad wt(\mathrm{Enc}_d(\vec{m})) \geq n_d - t_d \tag{7}$$

Thus, the relative minimum distance that C_d can achieve is

$$\Delta(C_d) = \frac{\min_{\vec{c} \neq \vec{0}, \vec{c} \in C_d} wt(\vec{c})}{n_d} = \frac{n_d - t_d}{n_d} = 1 - \frac{t_d}{n_d} \tag{3}$$

The result in equation (1) is derived from equation (3). The remaining task is to analyze what t_d equals, that is, how many zero components a codeword can have after encoding any non-zero message.

Utilizing Induction

Using the powerful tool of induction, we analyze t_d. Assume that with overwhelming probability (based on the choice of diagonal matrices $T_0, \ldots, T_{i-1}$), for any non-zero message $\vec{m} \in \mathbb{F}^{k_i} \setminus \{0^{k_i}\}$, the encoded $\mathrm{Enc}_i(\vec{m})$ has at most t_i zero components. We analyze the case for $i+1$. For any non-zero message $\vec{m} = (\vec{m}_l, \vec{m}_r) \in \mathbb{F}^{2k_i}$,

$$\begin{aligned} \mathrm{Enc}_{i+1}(\vec{m}) &= (\vec{m}_l, \vec{m}_r) \begin{bmatrix} \mathbf{G}_i & \mathbf{G}_i \\ \mathbf{G}_i \cdot T_i & \mathbf{G}_i \cdot -T_i \end{bmatrix} \\ &= (\vec{m}_l \mathbf{G}_i + \vec{m}_l \mathbf{G}_i \cdot T_i, \vec{m}_l \mathbf{G}_i - \vec{m}_l \mathbf{G}_i \cdot T_i) \\ &= (\mathrm{Enc}_i(\vec{m}) + \mathrm{Enc}_i(\vec{m}) \circ \mathrm{diag}(T_i), \mathrm{Enc}_i(\vec{m}) - \mathrm{Enc}_i(\vec{m}) \circ \mathrm{diag}(T_i)) \\ &:= (\mathbf{M}_l \| \mathbf{M}_r) \end{aligned} \tag{8}$$

This means examining the number of zero components in the vector $(\mathbf{M}_l \| \mathbf{M}_r)$. Separating $\mathbf{M}_l$ and $\mathbf{M}_r$:

$$\begin{aligned} \mathbf{M}_l &= \mathrm{Enc}_i(\vec{m}_l) + \mathrm{Enc}_i(\vec{m}_r) \circ \mathrm{diag}(T_i) \\ \mathbf{M}_r &= \mathrm{Enc}_i(\vec{m}_l) - \mathrm{Enc}_i(\vec{m}_r) \circ \mathrm{diag}(T_i) \end{aligned} \tag{9}$$

Let $\mathbf{t} = \mathrm{diag}(T_i)$. For each $j \in [1, n_i]$, define $A_j = \mathrm{Enc}_i(\vec{m}_l)[j]$, $B_j = \mathrm{Enc}_i(\vec{m}_r)[j]$, and define a function:

$$f_j(x) = A_j + xB_j \tag{4}$$

If $f_j(\mathbf{t}[j]) = 0$ or $f_j(-\mathbf{t}[j]) = 0$, then $\mathbf{M}_l[j] = 0$ or $\mathbf{M}_r[j] = 0$, indicating a zero component in the encoded vector. Let's analyze whether $f_j(x)$ can be zero based on the values of A_j and B_j, considering the following cases:

	$A_j = 0$	$A_j \neq 0$
$B_j = 0$	$f_j(x) \equiv 0$	$f_j(x) = A_j \neq 0$
$B_j \neq 0$	$f_j(x) = xB_j$	$f_j(x) = A_j + xB_j$

First, consider the case where $A_j = B_j = 0$. Here, $f_j(x) \equiv 0$ for any x, meaning $\mathbf{M}_l[j] = 0$ and $\mathbf{M}_r[j] = 0$. Let $S \subseteq [n_i]$ denote such indices, and by the induction hypothesis, $|S| \leq t_i$. Define $m_{i+1}(S)$ as those non-zero messages that satisfy $A_j = B_j = 0$, i.e.,

$$S = \{j \in [1, n_i] : A_j = B_j = 0\}. \tag{10}$$

In this case, $\mathbf{M}_l[j] = \mathbf{M}_r[j] = 0$, resulting in $2|S|$ zero components in $(\mathbf{M}_l || \mathbf{M}_r)$.

Consider the second case, where $A_j \neq 0$ and $B_j = 0$. Here, $f_j(x) = A_j \neq 0$, so no zero components are found.

Next, consider the last row of the table where $B_j \neq 0$. In this case, the index j is certainly not in S. Define a subset $\urcorner S^* \subseteq \urcorner S$ such that

$$\urcorner S^* = \{j \in [1, n_i] \backslash S, B_j \neq 0\} \tag{11}$$

For each $j \in \urcorner S^*$, define a random variable

$$X_j = 1\{f_j(\mathbf{t}[j]) = 0\} + 1\{f_j(-\mathbf{t}[j]) = 0\} \tag{12}$$

where $1(\cdot)$ is an indicator function that equals 1 if the condition inside holds, and 0 otherwise. Thus, X_j indicates how many zero components exist at position j in $\mathbf{M}_l$ and $\mathbf{M}_r$, with possible values $\{0, 1, 2\}$. Notice that X_j is an independent Bernoulli trial because $\mathbf{t}[j]$ is uniformly sampled from $\mathbb{F}^\times$. Let $z_j \in \mathbb{F}^\times$ satisfy $f_j(z_j) = 0$. Then, when $\mathbf{t}[j] = z_j$, $1\{f_j(\mathbf{t}[j]) = 0\} = 1$ and when $\mathbf{t}[j] = -z_j$, $1\{f_j(-\mathbf{t}[j]) = 0\} = 1$. Analyzing the possible values of X_j:

1. $X_j = 2$: This implies $f_j(\mathbf{t}[j]) = 0$ and $f_j(-\mathbf{t}[j]) = 0$, which leads to $\mathbf{t}[j] = z_j = -z_j$. This would mean $z_j = 0$, which is impossible since $z_j \in \mathbb{F}^\times$.
2. $X_j = 1$: This implies either $f_j(\mathbf{t}[j]) = 0$ or $f_j(-\mathbf{t}[j]) = 0$, meaning $\mathbf{t}[j] = z_j$ or $\mathbf{t}[j] = -z_j$. The probability of this occurring is $\frac{2}{|\mathbb{F}|-1}$.
3. $X_j = 0$: This occurs when $\mathbf{t}[j] \neq z_j$ and $\mathbf{t}[j] \neq -z_j$, with probability $1 - \frac{2}{|\mathbb{F}|-1}$.

For all $j \in \urcorner S^*$, summing up all X_j gives the total number of zero components in $(\mathbf{M}_l || \mathbf{M}_r)$, denoted as $X = \sum_{j \in \urcorner S^*} X_j$.

Having analyzed all cases in the table, we obtain

$$\mathrm{nzero}(\mathrm{Enc}_{i+1}(\vec{m})) = 2|S| + X \tag{13}$$

Next, we analyze $|S|$ and X to show that for any non-zero message $\vec{m} \in \mathbb{F}^{2k_i} \backslash \{0^{2k_i}\}$, $\mathrm{Enc}_{i+1}(\vec{m})$ has at most t_{i+1} zero components with overwhelming probability. We analyze the probability that $\mathrm{Enc}_{i+1}(\vec{m})$ has at least $2t_i + l_i$ zero components:

$$
\begin{aligned}
\Pr[\mathrm{nzero}(\mathrm{Enc}_{i+1}(\vec{m})) \geq 2t_i + l_i] &= \Pr[2|S| + X \geq 2t_i + l_i] \\
&= \Pr[X \geq 2t_i + l_i - 2|S|] \\
&= \Pr[\sum_{j \in \neg S^*} X_j \geq 2t_i + l_i - 2|S|] \\
&\leq \sum_{j=2t_i+l_i-2|S|}^{|\neg S^*|} \binom{|\neg S^*|}{i} \cdot (\frac{2}{|\mathbb{F}|-1})^i \cdot (1 - \frac{2}{|\mathbb{F}|-1})^{|\neg S^*|-i} \\
&\quad \text{(By the binomial theorem, } \binom{|\neg S^*|}{i} \leq 2^{|\neg S^*|}) \\
&\leq |\neg S^*| \cdot 2^{|\neg S^*|} (\frac{2}{|\mathbb{F}|-1})^{2t_i+l_i-2|S|} \\
&\leq |\neg S| \cdot 2^{|\neg S|} \cdot (\frac{2}{|\mathbb{F}|-1})^{2t_i+l_i-2|S|} \quad (\neg S^* \subseteq \neg S) \\
&= |[1, n_i] \backslash S| \cdot 2^{|[1,n_i]\backslash S|} \cdot (\frac{2}{|\mathbb{F}|-1})^{2t_i+l_i-2|S|} \\
&= (n_i - |S|) \cdot 2^{n_i - |S|} \cdot (\frac{2}{|\mathbb{F}|-1})^{2t_i+l_i-2|S|} \\
&\quad \text{(Assume } |\mathbb{F}| \geq 2^{10}, \text{ then } \frac{2}{|\mathbb{F}|-1} \leq \frac{2.002}{|\mathbb{F}|}) \\
&\leq n_i \cdot 2^{n_i-|S|} \left(\frac{2.002}{|\mathbb{F}|}\right)^{2t_i+l_i-2|S|}
\end{aligned} \tag{14}
$$

We observe that for an index set $S \subseteq [1, n_i]$, if any set S is selected, each index $i \in [1, n_i]$ has two possibilities: to include it in S or not. Therefore, there are a total of 2^{n_i} possible selections for the set S. When we enumerate all possible sets S, the union of the resulting $m_{i+1}(S)$, denoted by $\cup_{S \subseteq [1,n_i]} m_{i+1}(S)$, can cover all messages in $\mathbb{F}^{k_{i+1}} = \mathbb{F}^{2k_i}$. Lemma 2 in paper [ZCF23] tells us that the size of the set $m_{i+1}(S)$ is at most $\mathbb{F}^{t_i-|S|}$. Thus, by iterating through all 2^{n_i} possible sets S, each set S contains at most $\mathbb{F}^{t_i-|S|}$ messages $\vec{m} \in \mathbb{F}^{2k_i}$. By combining all S and considering the bounds on the size of each S, we can conclude that when l_i is sufficiently large, that is, when $|\mathbb{F}|^{l_i} \gg 2^{n_i}$, the expression $n_i \cdot 2^{n_i-|S|} \left(\frac{2.002}{|\mathbb{F}|}\right)^{2t_i+l_i-2|S|}$ becomes sufficiently small. In this case, for any non-zero vector $\vec{m} \in \mathbb{F}^{2k_i}$, we have $\mathrm{nzero}(\mathrm{Enc}_{i+1}(\vec{m})) \leq 2t_i + l_i$, that is, $\mathrm{Enc}_{i+1}(\vec{m})$ contains at most $2t_i + l_i$ zero components.

The BaseFold paper [ZCF23] presents a more specific statement in the form of a theorem.

Theorem 1 [ZCF23, Theorem 2] Fix any finite field $\mathbb{F}$ with $|\mathbb{F}| \geq 2^{10}$, and let $\lambda \in \mathbb{N}$ be the security parameter. For a vector $\mathbf{v}$ with components in $\mathbb{F}$, let $\mathrm{nzero}(\mathbf{v})$ denote the number of zero components in $\mathbf{v}$. For any $d \in \mathbb{N}$, let D_d be a (c, k_0)-foldable distribution, and for each $i \leq d$, set $k_i = k_0 2^i$, $n_i = ck_i$. Then,

$$
\Pr_{(\mathbf{G}_0, \ldots, \mathbf{G}_d) \leftarrow D_d} \left[\exists \mathbf{m} \in \mathbb{F}^{k_d} \backslash \{\mathbf{0}\}, \mathrm{nzero}(\mathrm{Enc}_d(\mathbf{m})) \geq t_d\right] \leq d \cdot 2^{-\lambda} \tag{5}
$$

where $t_0 = k_0$ and for each $i \in [d]$, $t_i = 2t_{i-1} + l_i$, with

$$
l_i := \frac{2(d-1)\log n_0 + \lambda + 2.002 t_{d-1} + 0.6 n_d}{\log |\mathbb{F}| - 1.001}. \tag{15}
$$

Equation (5) indicates that the number of zero components in $\mathrm{Enc}_d(\mathbf{m})$ is bounded by t_d with negligible probability if exceeded. Given the iterative formula $t_i = 2t_{i-1} + l_i$, we can compute t_d through iterative summation. Consequently, the maximum relative number of zero components in C_d is $Z_{C_d} = \frac{t_d}{n_d}$, and calculating $1 - Z_{C_d}$ yields the minimum relative Hamming distance Δ_{C_d} of C_d, resulting in equation (1):

$$
1 - \left(\frac{\epsilon_{\mathbb{F}}^d}{c} + \frac{\epsilon_{\mathbb{F}}}{\log |\mathbb{F}|} \sum_{i=0}^{d} (\epsilon_{\mathbb{F}})^{d-i} \left(0.6 + \frac{2\log(n_i/2) + \lambda}{n_i}\right) \right). \tag{16}
$$

From the iterative formula $t_i = 2t_{i-1} + l_i$, we observe that as i increases, t_i grows by more than $2t_{i-1}$. Since the encoding length doubles each iteration, the relative maximum number of zero components increases, and therefore the minimum relative Hamming distance decreases. If Δ_{C_d} is sufficiently large, then through this iterative method, we obtain with overwhelming probability that $\Delta_{C_0} \geq \Delta_{C_1} \geq \ldots \geq \Delta_{C_d}$. From $i = d$ to $i = 0$, this encoding method does not decrease Δ_{C_d}. In the IOPP protocol, if the initial minimum relative Hamming distance is large, then Δ_{C_0} remains large with overwhelming probability, which plays a significant role in analyzing the soundness of the IOPP.

References

- [ZCF23] Hadas Zeilberger, Binyi Chen, and Ben Fisch. "BaseFold: efficient field-agnostic polynomial commitment schemes from foldable codes." Annual International Cryptology Conference. Cham: Springer Nature Switzerland, 2024.
- [GJX15] Venkatesan Guruswami, Lingfei Jin, and Chaoping Xing. “Efficiently List-Decodable Punctured Reed-Muller Codes”. In: IEEE Transactions on Information Theory 63 (2015), pp. 4317–4324. url: https://api.semanticscholar.org/CorpusID: 14176561.