Enhanced pqsigRM: Code-Based Digital Signature Scheme with Short Signature and Fast Verification for Post-Quantum Cryptography

Jinkyu Cho[1], Jong-Seon No[1], Yongwoo Lee[2], Young-Sik Kim[3], and Zahyun Koo[4]

[1] Seoul National University, South Korea
jsno@snu.ac.kr, jgjo114@gmail.com
[2] Inha University, South Korea
yongwoo@inha.ac.kr
[3] Chosun University, South Korea
iamyskim@chosun.ac.kr
[4] Samsung Electronics, South Korea
bravokoo@gmail.com

Abstract. We present a novel code-based digital signature scheme, called Enhanced pqsigRM for post-quantum cryptography (PQC). This scheme is based on modified Reed–Muller (RM) codes, which modified RM codes with several security problems. Enhanced pqsigRM is a strengthened version of pqsigRM, which was submitted to NIST PQC standardization in round 1. The proposed scheme has the advantage of short signature size, fast verification cycles. For 128 bits of classical security, the signature size of the proposed scheme is 1032 bytes, which corresponds to 0.42 times that of Crystals-Dilithium, and the number of median verification cycles is 235,656, which is smaller than that of Crystals-Dilithium. Also, we use public codes, called modified RM codes, that are more difficult to distinguish from random codes. We use $(U, U+V)$-codes with high-dimensional hull to make these. Using modified RM codes, the proposed signature scheme resists various known attacks on RM-code-based cryptography. The proposed decoder samples from coset elements with small Hamming weight for any given syndrome and efficiently finds such elements.

Keywords: Code-based cryptography, digital signatures, error correction codes, post-quantum cryptography (PQC), Reed-Muller (RM) codes.

1 Introduction

Courtois, Finiasz, and Sendrier (CFS) proposed a signature scheme using high-rate Goppa codes in 2001 [1]. However, this scheme, so-called CFS signature scheme, has certain drawbacks in terms of scaling of the parameters and a lack of existential unforgeability under adaptive chosen message attacks (EUF-CMA). Further, its error correction capability t has to be small because the signing time depends on $t!$. The public key size of the CFS scheme is $(n-k)n = tm2^m$ and it is known that decoding attacks require $A = 2^{tm/2}$ operations. Thus the decoding attack complexity A is only a polynomial function of the key size with small power, that is, $A \approx \text{keysize}^{t/2}$. Therefore, because t should be kept as a relatively small value of up to 12 to reduce successful signing time, and we need to significantly increase the key size itself for higher security. Also, with a small t, the rate of Goppa codes is high. The parity check matrix of high-rate Goppa codes can be distinguished from a random matrix and thus the CFS signature scheme is insecure under the EUF-CMA [2].

In this submission, we replace the Goppa codes with the RM codes in the CFS signature scheme. RM codes can use complete decoding using well-known and efficient recursive decoding, called closest coset decoding [3], [4], that is, for a given received vector, the closest codeword can be found. The closest coset decoding method does not guarantee the exact error correction but finds an error vector (coset leader in the standard array) corresponding to the syndrome. However, the exact error correction is not essential for signing in code-based signature schemes, but we need to find the error vector with the smallest Hamming weight in the coset corresponding to the syndrome. In this respect, the RM code-based signature scheme can be considered as a solution to the small t constrained problem of the Goppa code-based signature scheme.

However, the simple replacement of Goppa codes with RM codes in the CFS signature scheme results in vulnerability to several attacks. The RM code-based McEliece cryptosystem is insecure under Minder–Shokrollahi's attack [5] and Chizhov–Borodin's attack [6]. With these two attacks, the private keys S, G, and Q can be revealed from the public key $G' = SGQ$, where G is a generator matrix and S and Q are a scrambling matrix and a permutation matrix, respectively. The above-mentioned attacks can be similarly applied to the RM code-based signature scheme. It is shown here that the proposed scheme is secure against these attacks.

We propose a new code-based signature scheme by using modified RM codes, called Enhanced pqsigRM. We first partially permute the original RM codes and proceed with three more modifications, which are replacing some parts of the code, appending random rows, and padding a dual code's codeword. For now, we propose one parameter set of Enhanced pqsigRM, that is, `Enh-pqsigRM-613` constructed by RM(6,13) for 128 bit-security. The proposed signature scheme is an improvement of pqsigRM [7] submitted to NIST for PQC standardization round 1, and it resolves the weaknesses of early versions of pqsigRM by modifying the public codes. Moreover, we ensure the indistinguishability of the public codes of the proposed signature scheme. Further, it can compromise the security level by adjusting the allowable maximum Hamming weight of error vectors, called the error weight parameter w. Our proposed scheme has the advantages of a small signature size, fast verification cycles. It is also proved that the proposed Enhanced pqsigRM is EUF-CMA secure. For 128 bits of classical security, the signature size of the proposed signature scheme is 1032 bytes, which corresponds to 0.42 times that of Crystals-Dilithium, and the number of median verification cycles is 235,656, which is smaller than that of Crystals-Dilithium.

1.1 Design Rationale

We introduce a new signature scheme, called Enhanced pqsigRM, based on modified RM codes with partial permutation as well as row appending and replacement in the generator matrix. For any given syndrome, an error vector with a small Hamming weight can be obtained. The proposed signature scheme resists all known attacks against cryptosystems based on the original RM codes. Using modified RM codes, we improve the security problems and indistinguishability of public codes. Assuming indistinguishability and the hardness of DOOM with a high-dimensional hull, we also achieve the EUF-CMA security of the proposed signature scheme.

1.2 Advantages and Limitations

Enhanced pqsigRM signature scheme has advantages in signature size. it has a relatively small signature size compared with the other digital signatures of NIST PQC finalist algorithms and code-based signatures. Also, it has a very short verification time for 128-bit security. The limitation of this scheme is the relatively large public key size. Since the codes in Enhanced pqsigRM do not have a structure such as quasi-cyclic, the key size of the public key is $(n-k) \times k$.

For 128 bits of classical security, the signature size of the proposed signature scheme is 1032 bytes, which corresponds to 0.42 times that of Crystals-Dilithium, and the number of median verification cycles is 235,656, which is smaller than that of Crystals-Dilithium.

2 Preliminaries

2.1 CFS Signature Scheme

CFS signature scheme is an algorithm that applies the full domain hash (FDH) methodology to the Niederreiter cryptosystem. It is based on Goppa codes, as McEliece public key encryption scheme. As described in Algorithm 1, the signing process iterates until a decodable syndrome is obtained. The probability of decoding a given random syndrome is $\frac{\sum_{i=0}^{t}\binom{n}{i}}{2^{n-k}} \simeq \frac{1}{t!}$. Hence, the error correction capability $t = \frac{n-k}{\log n}$ should be sufficiently small to reduce the number of iterations. Thus, high-rate Goppa codes should be used. Regarding the key size, the complexity of the decoding attack on the CFS signature scheme is known to be a small power of the key size, namely, $\approx \text{keysize}^{t/2}$. Hence, the key size should be fairly large to meet a certain security level. In summary, the original CFS signature scheme is insecure and inefficient due to using of Goppa codes.

Algorithm 1 CFS signature scheme [1]

```
Key generation:
    $\mathbf{H}$ is the parity check matrix of an $(n, k)$ Goppa code
    The error correction capability $t$ is $\frac{n-k}{\log n}$
    $\mathbf{S}$ and $\mathbf{Q}$ are an $(n-k) \times (n-k)$ scrambler matrix and $n \times n$ permutation matrix, respectively
    Secret key: $\mathbf{H}, \mathbf{S},$ and $\mathbf{Q}$
    Public key: $\mathbf{H}' \leftarrow \mathbf{SHQ}$
Signing:
    $\mathbf{m}$ is a message to be signed
    $i \leftarrow 1$
    Do
        $i \leftarrow\ i + 1$
        Find syndrome $\mathbf{s} \leftarrow h(h(\mathbf{m})|i)$
        Compute $\mathbf{s}' \leftarrow \mathbf{S}^{-1}\mathbf{s}$
    Until a decodable syndrome $\mathbf{s}'$ is found
    Find an error vector satisfying $\mathbf{He}'^T \leftarrow \mathbf{s}'$
    * Compute $\mathbf{e}^T \leftarrow \mathbf{Q}^{-1}\mathbf{e}'^T$, and then the signature is $(\mathbf{m}, \mathbf{e}, i)$
Verification:
    Check $\mathrm{wt}(\mathbf{e}) \leq t$ and $\mathbf{H}'\mathbf{e}^T = h(h(\mathbf{m})|i)$
    If true, then return ACCEPT; else, return REJECT
```

2.2 Reed–Muller Codes and Recursive Decoding

RM codes were introduced by Reed and Muller [8, 9] and its decoding algorithm, so-called recursive decoding, was proposed in [4]. $\mathrm{RM}_{(r,m)}$ is a linear binary $(n = 2^m, k = \sum_{i=0}^{r} \binom{m}{i})$ codes, where r and m are integers. $\mathrm{RM}_{(r,m)}$ is defined as $\mathrm{RM}_{(r,m)} := \{(\mathbf{u}|\mathbf{u}+\mathbf{v})|\mathbf{u} \in \mathrm{RM}_{(r,m-1)}, \mathbf{v} \in \mathrm{RM}_{(r-1,m-1)}\}$, where $\mathrm{RM}_{(0,m)} := \{(0, \ldots, 0), (1, \ldots, 1)\}$ with code length 2^m and $\mathrm{RM}_{(m,m)} := \mathbb{F}_2^{2^m}$. In other words, we can make a recursive structure by Plotkin's construction, and its generator matrix is given by

$$\mathbf{G}_{(r,m)} = \begin{bmatrix} \mathbf{G}_{(r,m-1)} & \mathbf{G}_{(r,m-1)} \\ \mathbf{0} & \mathbf{G}_{(r-1,m-1)} \end{bmatrix},$$

where $\mathbf{G}_{(r,m)}$ is the generator matrix of $\mathrm{RM}_{(r,m)}$.

Recursive decoding is a soft-decision decoding algorithm that depends on the recursive structure of the RM codes; it is described in detail in Algorithm 2, where $\mathbf{y}' \cdot \mathbf{y}''$ denotes the component-wise multiplication of the vectors $\mathbf{y}'$ and $\mathbf{y}''$. In recursive decoding, a binary symbol $a \in \{0, 1\}$ is mapped onto $(-1)^a$, and it is assumed that all codewords belong to $\{-1, 1\}^n$.

First, $\mathbf{y}''$ (the second half of the received vector $\mathbf{y}$) is a component wisely multiplied by $\mathbf{y}'$ (the first half of the received vector). Then, a codeword from $\mathrm{RM}_{(r,m-1)}$ (i.e., $\mathbf{u}$) is removed from $\mathbf{y}''$ as it is both in $\mathbf{y}'$ and $\mathbf{y}''$, and then only $\mathbf{v}$ and the error vector remain. This is regarded as a codeword of $\mathrm{RM}_{(r-1,m-1)}$ added to an error vector and is referred to as $\hat{\mathbf{v}}$. Using $\hat{\mathbf{v}}$, we can remove the codeword of $\mathrm{RM}_{(r-1,m-1)}$ from the second half of the received vector. $\mathbf{y}'$ is then added to $\mathbf{y}'' \cdot \hat{\mathbf{v}}$, and the sum is divided by 2. This is regarded as a codeword of $\mathrm{RM}_{(r,m-1)}$ added to the error vector, and then decoding is performed. Recursively, the received vector is further divided into sub-vectors of length $n/4$, $n/8$, etc. Finally, we reach $\mathrm{RM}_{(m,m)}$ or $\mathrm{RM}_{(0,m)}$, then the division terminates and the minimum distance (MD) decoding of $\mathrm{RM}_{(m,m)}$ or $\mathrm{RM}_{(0,m)}$, which is trivial, is performed. The decoding for the entire code is performed by reconstructing these results into $(U, U + V)$ form.

Algorithm 2 Recursive decoding of RM code [4]

```
function RecursiveDecoding($\mathbf{y}, r, m$)
    if $r = 0$ then
        Perform MD decoding on RM($0, m$)
    else if $r = m$ then
        Perform MD decoding on RM($r, r$)
    else
        $(\mathbf{y}'|\mathbf{y}'') \leftarrow \mathbf{y}$
        $\mathbf{y}^{\mathbf{v}} = \mathbf{y}' \cdot \mathbf{y}''$
        $\hat{\mathbf{v}} \leftarrow$ RecursiveDecoding($\mathbf{y}^{\mathbf{v}}, r-1, m-1$)
        $\mathbf{y}^{\mathbf{u}} \leftarrow (\mathbf{y}' + \mathbf{y}'' \cdot \hat{\mathbf{v}})/2$
        $\hat{\mathbf{u}} \leftarrow$ RecursiveDecoding($\mathbf{y}^{\mathbf{u}}, r, m-1$)
        Output $(\hat{\mathbf{u}}|\hat{\mathbf{u}} \cdot \hat{\mathbf{v}})$
    end if
end function
```

3 Specification

3.1 Basic Notation

A vector is denoted in boldface in the form of a column vector. $(\mathbf{x}_0|\mathbf{x}_1)$ denotes the concatenation of two vectors $\mathbf{x}_0$ and $\mathbf{x}_1$. For example, $h(\mathbf{m}|r)$ means the hash function h with input $(\mathbf{m}|r)$, where $(\mathbf{m}|r)$ represents the concatenation of the binary representation of vector $\mathbf{m}$ and a random value r. Matrices are denoted by a boldfaced capital letter, for example, $\mathbf{A}$. Matrix multiplication is denoted by $\cdot$ or can be omitted when it is unnecessary. Codes and probability distributions are denoted in calligraphic fonts, for example, $\mathcal{C}$, and it can be distinguished by context. $\mathbf{x}^{\sigma}$ denotes that a vector $\mathbf{x}$ is permuted by a permutation σ, for example, $\mathbf{x}^{\sigma} = (x_1, x_3, x_2, x_0)$, where $\mathbf{x} = (x_0, x_1, x_2, x_3)$ and $\sigma = (1, 3, 2, 0)$.

3.2 Parameter Space

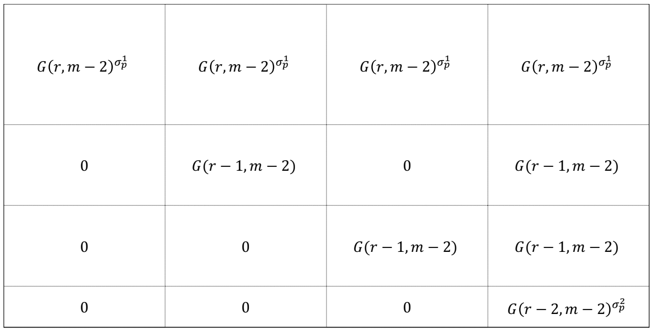
We propose a new code-based digital signature scheme, called Enhanced pqsigRM. Each operation of Enhanced pqsigRM has six parameters: (r, m) are positive integers of parameters of RM code, p is the number of columns that are partially permuted, w is the Hamming weight of signature, k_{rep} is the number of replacing rows, and k_{app} is the number of appending rows.

3.3 Constructing Modified RM Codes

1) Partial permutation of generator matrix of RM code:

For a code $\mathcal{C}$, we define its hull by the intersection of the code and its dual, in other words, $hull(\mathcal{C}) = \mathcal{C} \cap \mathcal{C}^{\perp}$. The proposed $(U, U+V)$-code is designed to have a high-dimensional hull, where $dim(U^{\perp} \cap V)$, dimension of $U^{\perp} \cap V$, is large. In general, for a $(U, U+V)$-code $\mathcal{C}$, a codeword $(\mathbf{u}|\mathbf{u}+\mathbf{v}) \in hull(\mathcal{C})$ satisfies $\mathbf{v} = \mathbf{u}^{\perp}$ and $\mathbf{u} + \mathbf{v} = \mathbf{v}^{\perp}$, where $\mathbf{u} \in U$ and $\mathbf{v} \in V$. Hence, when $U^{\perp} \cap V = \{\mathbf{0}\}$, $hull(\mathcal{C})$

has only $(\mathbf{u}|\mathbf{u})$ codewords, and this may reveal the secret key. To avoid this, the proposed code is designed so that $dim(U^{\perp} \cap V)$ is large.



$G(r,m-2)^{\sigma_p^1}$	$G(r,m-2)^{\sigma_p^1}$	$G(r,m-2)^{\sigma_p^1}$	$G(r,m-2)^{\sigma_p^1}$
0	$G(r-1,m-2)$	0	$G(r-1,m-2)$
0	0	$G(r-1,m-2)$	$G(r-1,m-2)$
0	0	0	$G(r-2,m-2)^{\sigma_p^2}$

Fig. 1. Partially permuted RM code's generator matrix.

First, we construct the generator matrix $\mathbf{G}_{(r,m)}$ of an RM code and then permute its submatrices. An example is shown in Figure 1, where σ_p^1 and σ_p^2 denote two independent partial permutations that randomly permute only p out of $n/4$ columns. To generate σ_p^1 and σ_p^2, p column indices are randomly selected from the index set $\{0, 1, \ldots, n/4-1\}$, and the selected indices are randomly permuted, whereas the others are not. Then, σ_p^1 is used to permute the submatrices corresponding to $\mathbf{G}_{(r,m-2)}$'s in the first $dim(\mathrm{RM}_{(r,m-2)})$ rows, and σ_p^2 is used to permute the submatrix corresponding to $\mathbf{G}_{(r-2,m-2)}$ in the last $dim(\mathrm{RM}_{(r-2,m-2)})$ rows, as shown in Figure 1. The codes generated by the generator matrix in Figure 1 are called partially permuted RM codes. It should be noted that, unlike in the case of code-based cryptographic algorithms, we permute submatrices of the generator matrix rather than the entire matrix here. We note that the entire matrix should also be permuted to design a signature scheme.

$dim(U^{\perp} \cap V)$ is large for the following reasons. Let $\mathbf{G}_U$ and $\mathbf{G}_V$ denote the generator matrices of U and V, respectively:

$$\mathbf{G}_U = \begin{bmatrix} \mathbf{G}_{(r,m-2)}^{\sigma_p^1} & \mathbf{G}_{(r,m-2)}^{\sigma_p^1} \\ \mathbf{0} & \mathbf{G}_{(r-1,m-2)} \end{bmatrix},$$

$$\mathbf{G}_V = \begin{bmatrix} \mathbf{G}_{(r-1,m-2)} & \mathbf{G}_{(r-1,m-2)} \\ \mathbf{0} & \mathbf{G}_{(r-2,m-2)}^{\sigma_p^2} \end{bmatrix}.$$

Then, the generator matrix of the dual code of U is

$$\mathbf{G}_U^{\perp} = \begin{bmatrix} \mathbf{G}_{(r,m-2)}^{\perp\sigma_p^1} & \mathbf{0} \\ \mathbf{G}_{(r-1,m-2)}^{\perp} & \mathbf{G}_{(r-1,m-2)}^{\perp} \end{bmatrix}.$$

Thus, $U^{\perp} \cap V$ has a subcode that is the intersection of the codewords generated by $\left[\mathbf{G}_{(r-1,m-2)}\ \mathbf{G}_{(r-1,m-2)}\right]$ and the codewords generated by $\left[\mathbf{G}^{\perp}_{(r-1,m-2)}\ \mathbf{G}^{\perp}_{(r-1,m-2)}\right]$. Its dimension is $\min(dim(\mathrm{RM}_{(r-1,m-2)}, dim(\mathrm{RM}_{(m-r-2,m-2)}))$, as the dual of $\mathrm{RM}_{(r,m)}$ is equal to $\mathrm{RM}_{(m-r-1,m)}$ and $\mathrm{RM}_{(r',m)} \subseteq \mathrm{RM}_{(r,m)}$, where $r' \leq r$.

2) Modification by replacing, appending, and padding:

With the partially permuted RM codes, the received vector and the syndrome have the same parity, causing the signature leak. Thus, the generator matrix in Figure 1 should be further modified.

That is, some rows are replaced with repetitions of random codewords, and random rows are appended to the generator matrix. Considering $\mathbf{G}_U$, it is also a $(U, U+V)$-code, which can similarly be divided into (permuted) $(U, U+V)$-codes. By repeating this process 2^{m-r} times, the rows of the partially permuted RM code consist of the 2^{m-r} repeated generator matrices of $\mathrm{RM}_{(r,r)}$, which are $2^r \times 2^r$ identity matrices. Then, $\mathrm{RM}_{(r,r)}$ is replaced by a repeated random $(2^r, k_{rep})$ code such that its dual code has at least one non-zero codeword with an odd hamming weight.

We now append random independent rows to the generator matrix. One row to be appended is a random codeword of the dual code. This should be independent of the existing rows; i.e., it should not belong to the hull of the code. Furthermore, it should be verified that the hull has codewords with Hamming weight that is not a multiple of four as a result of appending this row. The others are k_{app} random independent vectors including at least one vector of odd Hamming weight. These k_{app} vectors are independent of the partially permuted RM codes and independent of each other.

After all these modifications, the resulting code is called a modified RM code. An example of its generator matrix is given in Figure 2. We use k_{rep} as $2^r - 2$, which means we erase two rows. Then, we append two random rows and one dual code's codeword. Thus, the dimension of modified RM codes is larger than the original RM codes by 1.

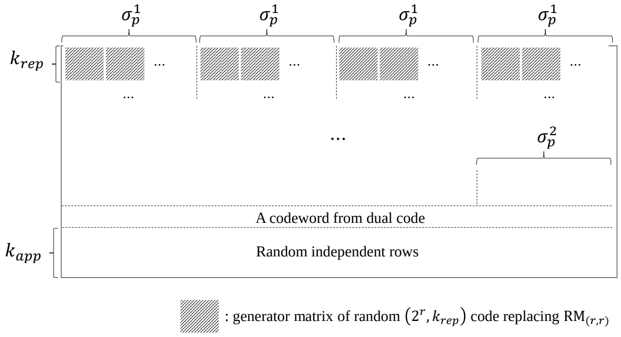

σ_p^1
σ_p^1
σ_p^1
σ_p^1
k_{rep}
...
σ_p^2
A codeword from dual code
k_{app}
Random independent rows
▨ : generator matrix of random $(2^r, k_{rep})$ code replacing $\mathrm{RM}_{(r,r)}$

Fig. 2. Modified RM code's generator matrix $G_{\mathcal{M}}$ for the proposed signature scheme.

3) Decoding of modified RM codes:

Unlike the Niederreiter cryptosystem and CFS signature scheme, it is required to find an error vector whose Hamming weight is larger than the error correction capability. Hence, there may exist several solutions $\mathbf{e}$ satisfying $\mathbf{He}^T = \mathbf{s}^T$ and $\mathrm{wt}(\mathbf{e}) \leq w$ for a given syndrome $\mathbf{s}$. Such decoding can be achieved by the modified Prange decoder using the $(U, U+V)$ structure, as in the signature schemes in [10, 11]. However, a new decoder is proposed that uses the recursive structure of the subcode of modified RM codes and it achieves better performance than the modified Prange decoder. In other words, it finds error vectors whose Hamming weights are less than the result in [10]. This results in the smaller parameters, considering attacks as in [12].

In addition to the decoding performance, a major difference between the proposed decoder and the modified Prange decoder is their input. The input of the modified Prange decoder used in [10] and [11] is a syndrome vector. In contrast, the input of the proposed decoder is an n dimensional vector $\mathbf{r}$ satisfying $\mathbf{Hr}^T = \mathbf{s}$, which is called a received vector in coding theory, and the decoder outputs codewords close to the received vector. An error vector with a small Hamming weight is obtained by subtracting the output from the received vector. Even if two different received vectors in the same coset are given, the proposed decoder can return different outputs. Besides, as the input of the decoder is a random received vector, decoding can be performed even if random rows are appended to the generator matrix.

As stated in the previous section, random rows (one from the dual code and the others being k_{app} independent random vectors) are appended to the generator matrix of the partially permuted RM codes. Let $\mathcal{C}_{app}$ be the code spanned by the added $k_{app}+1$ rows. The number of codewords increases by $2^{k_{app}+1}$ times when rows are appended by adding codewords of $\mathcal{C}_{app}$ to each $(U, U+V)$-codeword. Choosing a codeword of $\mathcal{C}_{app}$ (including $\mathbf{0}$), subtracting it from the received vector $\mathbf{r}$, decoding it, and adding the subtracted codewords back is the decoding process when rows are appended. Thus, the code is decodable even if arbitrary random codes are appended to its generator matrix.

Hence, it suffices to explain the decoding algorithm for the $(U, U+V)$-subcode of a modified RM code. This decoding follows the recursive decoding of RM codes [4]. The difference is the partial permutation and the replacement of $\mathrm{RM}_{(r,r)}$. Considering the decoding proposed in [4], we have $\mathbf{c} = (\mathbf{u}|\mathbf{u}+\mathbf{v})$ for all $\mathbf{c} \in \mathrm{RM}_{(r,m)}$, where $\mathbf{u} \in \mathrm{RM}_{(r,m-1)}$ and $\mathbf{v} \in \mathrm{RM}_{(r-1,m-1)}$. $\mathrm{RM}_{(r,m-1)}$ and $\mathrm{RM}_{(r-1,m-1)}$ are also $(U, U+V)$-codes, except for $r = 0$ or $r = m$. Here, if the code corresponding to $\mathbf{u}$ or $\mathbf{v}$ is replaced with a code other than the RM code and decoding the replaced code can be performed appropriately, the entire code $\mathbf{c}$ can also be decoded [3].

When the subcode of the RM code is replaced with its permutation, the entire code can also be decoded by slightly modifying the recursive decoding. Moreover, no decoding failure occurs because the recursion eventually reaches $\mathrm{RM}_{(0,m')}$, $\mathrm{RM}_{(r',r')}$, or the $(2^r, k_{rep})$ code to replace $\mathrm{RM}_{(r,r)}$ and there exists polynomial-time MD decoder for these codes. Even the $(2^r, k_{rep})$ random code

Algorithm 3 Decoding for modified RM code

```
function Decode($\mathbf{s}$; $\mathbf{H}$)
    $\mathbf{r} \leftarrow$ Prange($\mathbf{H}, \mathbf{s}$)
    while True do
        $\mathbf{r} \leftarrow \mathbf{r}+$ random codeword
        $\mathbf{c} \leftarrow$ ModDec($\mathbf{r}, r, M$)
        if $\mathrm{wt}(\mathbf{r}+\mathbf{c}) \leq w$ then
            Output $\mathbf{r}+\mathbf{c}$
        end if
    end while
end function

function ModDec($\mathbf{y}, r, M$)
    $\mathbf{y} \leftarrow \mathbf{y}^{\sigma^{-1}}$
    if $r = 0$ then
        Output MD decoding on RM$(0, m)$
    else if $r = m$ then
        Output MD decoding on RM$(r, r)$
         or replaced $(2^r, k_{rep})$ code
    else
        $(\mathbf{y}'|\mathbf{y}'') \leftarrow \mathbf{y}$
        $\mathbf{y}^{\mathbf{v}} = \mathbf{y}' \cdot \mathbf{y}''$
        $\hat{\mathbf{v}} \leftarrow$ ModDec($\mathbf{y}^{\mathbf{v}}, r-1, m-1$)
        $\mathbf{y}^{\mathbf{u}} \leftarrow (\mathbf{y}' + \mathbf{y}'' \cdot \hat{\mathbf{v}})/2$
        $\hat{\mathbf{u}} \leftarrow$ ModDec($\mathbf{y}^{\mathbf{u}}, r, m-1$)
        $\mathbf{y} \leftarrow (\hat{\mathbf{u}}|\hat{\mathbf{u}} \cdot \hat{\mathbf{v}})$
    end if
    Output $\mathbf{y}^{\sigma}$
end function
```

*σ is σ_p^1 or σ_p^2 for permuted block and identity, otherwise.

is MD decodable in constant time because it is a small code. To handle partial permutations, when the code is decodable, it uses the fact that the permutation is always decodable if the permutation is known.

In general, the output distribution of decoding is crucial for security. Thus, we also propose a randomized decoding method, the output of which is almost uniformly distributed. Using the algorithm described above, a random decoder can easily be designed. Algorithm 3 summarizes the randomized decoding. It is easy to find a received vector (regardless of its Hamming weight) for any given syndrome; a coset element corresponding to the syndrome is randomly selected. This is given to the decoder as an input. Finally, the decoder finds a different error vector with a small Hamming weight for different inputs.

3.4 Generation of Digital Signatures

Then we use the modified RM codes in the process of the signature scheme of *Enhanced pqsigRM* as in Algorithm 4. It is composed of key generation, signing, and verification processes as below.

1) Key generation : Let $G_{\mathcal{M}}$ be the modified RM code's generator matrix of RM(r, m) in Fig. 2. It has a code length of n and dimension k. The dual matrix of $G_{\mathcal{M}}$ becomes the parity check matrix, which is denoted by $H_{\mathcal{M}}$.

Let Q be an $n \times n$ permutation matrix, which is randomly chosen. Q is generated by a random shuffling algorithm (such as Knuth's shuffling algorithm [13]) using random numbers. The random numbers are made by a random number generator based on AES-256 (shortly, RNG-AES-256).

Then, we compute $H_{sys} = S_{sys} H_{\mathcal{M}} Q$. S_{sys} is a unique matrix, which makes $H_{\mathcal{M}} Q$ to be a systematic form. H_{sys} can be expressed as $(I|T)$ and T becomes the public key, which is an $(n-k) \times k$ matrix. The secret keys are Q, σ_p^1, σ_p^2, $k_{rep} \times 2^r$ (repeated) replacing codes, $k_{app} \times n$ appending codes, and $1 \times n$ padding dual code codeword.

2) Signing : For a given message M, choose random integer i generated by RNG-AES-256. Using the hash function h, the syndrome $s = h(M|i)$ is generated, which is similar to that of the CFS signature scheme. Unlike the CFS signature scheme, we use the hash function once, instead of twice. We use SHAKE-128 as a hash function. Then we make s' from s by multiplying the inverse of S_{sys}. Then we use the decoding algorithm of modified RM codes (Algorithm 3) to get e' from s'. Finally, e is generated from e' by multiplying the inverse of Q. The signature is composed of message M, error e, and counter i.

3) Verification : For verification, we check whether the computation from e and the computation from M are the same or not. That means we check two conditions, which are wt$(e) \leq w$ and $H_{sys} e^T = h(M|i)$. If these are satisfied, we return ACCEPT. If not, we return REJECT.

3.5 Parameter Sets

Parameter set Enh-pqsigRM-613 : Uses RM code RM(6,13) with $w = 1370$ and $p = 572$ (128-bit security).

The sizes of the public key and signature are given in Table 1. Compared with the NIST PQC finalist signature schemes [14–16], our scheme has the smallest signature size except for Falcon. Also, we compare these with the other code-based signature schemes [17, 11, 18, 19] in Table 2. Enhanced pqsigRM has the smallest signature size among these. Also, it has a smaller public key size than Wave. Durandal has an extremely small public key, however, its security relies on the security rank metric decoding problem. Classic McEliece is a key encapsulation mechanism (KEM), however, we brought this to compare with our scheme. If Classic McEliece can be acceptable as a NIST PQC candidate, our scheme is also acceptable regarding the public key size.

Furthermore, the specific parameter sets of *Enhanced pqsigRM* are as in Table 3.

```
Algorithm 4 Signature scheme of Enhanced pqsigRM
Key Generation :
    $\mathbf{G}_{\mathcal{M}}$: $k \times n$ generator matrix of modified RM codes
    $\mathbf{H}_{\mathcal{M}}$: $(n-k) \times n$ parity check matrix of modified RM codes
    $\mathbf{Q} \overset{\$}{\leftarrow} F_2^{n \times n}$
    $\mathbf{H_{sys}} = (\mathbf{I}|\mathbf{T}) \leftarrow \mathbf{S_{sys}}\mathbf{H}_{\mathcal{M}}\mathbf{Q}$
    Public key: $\mathbf{T}$
    Secret key: $\mathbf{Q}$, $\sigma_p^1, \sigma_p^2$, $k_{rep} \times 2^r$ (repeated) replacing codes, $k_{app} \times n$ appending
    codes, and $1 \times n$ padding dual code codeword

Signing :
    $M$: Message, $i \leftarrow \{0,1\}^{\lambda_0}$: Counter
    $\mathbf{s} \leftarrow h(M|i)$: Syndrome
    $\mathbf{s}'^T \leftarrow \mathbf{S_{sys}}^{-1}\mathbf{s}^T$
    $\mathbf{e}' \leftarrow \textsc{Decode}(\mathbf{s}'; \mathbf{H}_{\mathcal{M}})$
    $\mathbf{e}^T \leftarrow \mathbf{Q}^{-1}\mathbf{e}'^T$
    Signature: $(M, \mathbf{e}, \mathbf{i})$

Verification :
    If $wt(\mathbf{e}) \leq w$ and $\mathbf{H_{sys}}\mathbf{e}^{\mathbf{T}} = h(M|i)$,
        return ACCEPT
    Else, return REJECT

*$h$: hash function SHAKE-128
*DECODE: Decoding algorithm of modified RM codes
*$wt(a)$: Hamming weight of a vector $a$
*$w$: error correcting capability of modified RM codes
```

Table 1. Public key and signature sizes of *Enhanced pqsigRM* compared with the NIST PQC finalist signature schemes

Security	*Enhanced pqsigRM*		*Crystals-Dilithium*[14]		*Falcon*[15]		*Sphincs+*[16]	
	Public key(MB)	Signature (byte)	Public key(byte)	Signature (byte)	Public key(byte)	Signature (byte)	Public key(byte)	Signature (byte)
128	2.00	1,032	1,312	2,420	897	666	32	7,856

Table 2. Public key and signature sizes of *Enhanced pqsigRM* compared with other code-based signature schemes

Security	*Enhanced pqsigRM*		*Wave*[11]		*Durandal*[18]		*Classic McEliece*[19] (KEM)
	Public key(MB)	Signature (byte)	Public key(MB)	Signature (byte)	Public key(MB)	Signature (byte)	Public key (MB)
128	2.00	1,032	3.10	1,647	0.015	4,060	0.26

Table 3. Parameter sets for *Enhanced pqsigRM*

λ (security)	128
(r, m)	(6,13)
n	8192
k	4097
w	1370
p	$\geq$572
k_{rep}	62
k_{app}	2

4 Performance Analysis

4.1 Description of Platform

The following measurements are collected using a desktop computer with a CPU —i7-12700 CPU @ 2.10GHz—. Turbo Boost is disabled. This machine has 16GB of RAM. Benchmarks run on one core of the CPU. Since the signing algorithm is a probabilistic algorithm, the number of iterations at signing varies. The following result is the average of 100 experiments.

NIST said that the “NIST PQC Reference Platform” is “an Intel x64 running Windows or Linux and supporting the GCC compiler”. Our system is an x64 running Linux and supporting the GCC compiler. Beware, however, that different Intel CPUs can output different results.

4.2 Number of Cycles for Verification, Key Generation, and Signing

The following measurements are CPU cycler for running `Enh-pqsigRM-613` at —i7-12700 CPU @ 2.10GHz—. The measurements compared with the NIST PQC finalist algorithms are given in Table 4. The data of these are from the submission papers and these can be a little bit different because their implementation conditions are different [14–16]. However, these are almost the same as Crystals-Dilithium.

The verification CPU cycles of `Enh-pqsigRM-613` are 242,901 (average) and 235,656 (median). It is the smallest value among other NIST PQC finalists except for *Falcon*.

Table 4. Verification CPU cycles of *Enhanced pqsigRM* compared with the NIST PQC finalists

Security	Verification cycles			
	Enhanced pqsigRM	*Crystals-Dilithium*[14]	*Falcon*[15]	*Sphincs+*[16]
128	242,901	327,362	82,340	308,774

Furthermore, for key generation, these are 2,034,133,439 (average) and 2,038,358,872 (median). For signing, these are 2,232,288 (average) and 1,366,500 (median).

4.3 Memory Usage

`Enh-pqsigRM-613` takes 53,334,140 bytes of memory usage. Also, there is no memory leakage.

5 Design Rationale

5.1 Choosing Parameter Sets

The constraint here is that n is a power of two. We can numerically find the feasible ranges of w once n and k are determined. If the security level λ is achieved in this range, we accept the value; otherwise, we increase n. Considering decoding one out of many (DOOM) problem, which is explained in Section 6.2, a smaller value of w implies higher security. If w is so small that a large number of decoding iterations are required, we could reduce the partial permutation parameter p. p is at most $n/4$, and the characteristics of the codes are retained by lowering p to a certain degree. The method for obtaining the minimum values is described in the following subsection. The discussed state-of-the-art algorithm for DOOM is used as a basis for the parameters.

Regarding the key size, the public key is a parity check matrix given in the systematic form and requires $(n-k)k$ bits. The secret key includes matrix $\mathbf{Q}$, partial permutation σ_p^1, σ_p^2, $k_{rep} \times 2^r$ repeated replacing codes, $k_{app} \times n$ appending codes, and $1 \times n$ padding dual code codeword. $\mathbf{Q}$ is an $n \times n$ permutation matrix, which can be expressed with just a number. We use nm bits for $\mathbf{Q}$ because we need $\log_2 n$ bits to express a number and the number is from 0 to n. In the same way, σ_p^1 and σ_p^2 need $n(m-2)/2$ bits. The replacing codes, appending codes, and padding codes need $(2^r - 2) \times 2^r, k_{app} \times n$, and $1 \times n$ bits, respectively. Thus, the size of the secret key is $3nm/2 + k_{app}n + (2^r - 2)2^r$. It is 22,512 bytes for `Enh-pqsigRM-613`. On the other hand, I need $n + 64$ bits for the signature length. n is for the length of e, and 64 is for the size of a 64-bit integer i. The signature length is 1032 for `Enh-pqsigRM-613`. Moreover, $\mathbf{H}$ can be represented by $\sigma_p^1, \sigma_p^2, k_{rep} = 2^r - 2$ (the maximum value), and $k_{app} = 2$(the minimum value).

5.2 Statistical Analysis for Determining Number of Partial Permutations

The number p of columns permuted in the partial permutation varies from 0 to $n/4$. From numerical analysis, it is demonstrated that small values of p result in a low Hamming weight of the decoding output. However, it should be noted that when $p = 0$, the $(U, U + V)$ part of the modified RM codes becomes identical to the RM code except that $\mathrm{RM}_{(r,r)}$ is replaced. Hence, we propose the lower bound of p that does not affect the randomness of the hull.

Regarding the modified RM code, its hull overlaps with (but is not a subset of) the original RM code. If the hull is a subset of the original RM code, and its dimension is large, the codeword of the minimum Hamming weight of the original RM code may be included in the hull. Then, attacks such as the Minder–Shokrollahi attack may be applied using codewords with minimum Hamming weight. Therefore, to prevent attacks, the hull of the public code should not be a subset of the original RM code, and $hull(\mathcal{C}_{pub}) \setminus (\mathrm{RM}_{(r,m)}$ permuted by Q) should occupy a large portion of the hull, where $\mathcal{C}_{pub}$ denotes the public code, and $\setminus$ denotes the relative complement.

As the permutation Q is not important for determining the parameter p, we ignore it in this subsection, and the term permutation refers to the partial permutations σ_p^1 and σ_p^2. When $p = n/4$, which implies that σ_p^1 and σ_p^2 are full permutations, the average dimension of the hull and the dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ are given in Table 5. The values may slightly change according to the permutation.

If p is small, the Hamming weight of the errors decreases. Hence, the signing time can be reduced by using a partial permutation with p rather than a full permutation. The aim is to find a smaller value for p maintaining the dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ as large as that by the full permutation. It can be seen that the average of the dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ tends to increase as p increases, and it is saturated when p is above a certain value, as in Figure 3. Specifically, the dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ is saturated when p is approximately equal to the average dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ with full permutation. Hence, we determine p as 572.

Table 5. Average dimension of $hull(\mathcal{C}_{pub})$ and $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ with $p = n/4$

(r, m)	(6,13)
n	8192
k	4097
$dim(hull(\mathcal{C}_{pub}))$	2974
$dim(hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)})$	572

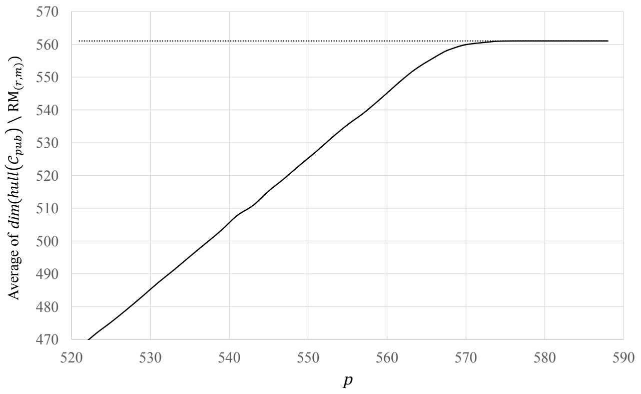


Fig. 3. Dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(6,13)}$ for 128-bit security parameters.

6 Security Analysis and Indistinguishability

6.1 RM Code Structure Attack

Minder-Shokrollahi's attack [5] and Chizhov-Borodin's attack [6] are well-known attacks for RM code-based cryptosystem, which decomposes the public key $H' = SHQ$ into the private keys S, H, and Q. In addition, square code attack [20] can also be applied to RM code-based cryptosystem with insertion. However, in the proposed scheme, because of the partial permutation, replacement, and appending codewords in the generator matrix, these attacks are not available.

6.2 Security Analysis

Decoding one out of many (DOOM) : The information set decoding is a brute-force attack method that finds an error vector $\mathbf{e}$ such that $\mathbf{He}^T = \mathbf{s}$ and $\mathrm{wt}(\mathbf{e}) \leq w$, where Stern improved the attack complexity in [21]. It has been extensively studied, and Dumer's algorithm [22] as well as more involved variants in [23, 24] have been proposed.

In the variants of the CFS signature scheme, there are several hash queries. Therefore, to launch a forgery attack, it suffices to find an error vector with a small Hamming weight for any of the syndromes. Hence, the decoding problem DOOM given below is adequate for a tight security proof. The usual FDH proof for existential forgery using syndrome decoding would require a work factor $\geq q_{\mathcal{H}} \cdot 2^{\lambda}$, where $q_{\mathcal{H}} \leq 2^{\lambda}$ is the number of hash queries. However, with DOOM, the work factor is required to be $\geq 2^{\lambda}$. Although the work factor of DOOM is greater than that of syndrome decoding, it provides tighter bounds for security.

Problem 1. (DOOM problem)

Instance: A parity check matrix $\mathbf{H} \in \mathbb{F}_2^{(n-k)\times n}$ of an (n,k) linear code, syndromes $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_q \in \mathbb{F}_2^{n-k}$, and an integer w.

Output: $(\mathbf{e}, i) \in \mathbb{F}_2^n \times [1,q]$ such that $\mathrm{wt}(\mathbf{e}) \leq w$ and $\mathbf{He}^T = \mathbf{s}_i^T$.

We consider the case in which the adversary has q instances and $M = \max(1, \binom{n}{w}/2^{n-k})$ solutions for each instance. Of course, in our case, w is not small, and thus M is $\binom{n}{w}/2^{n-k}$. In [12], the work factor of solving DOOM is given as

$$\mathrm{WF}_q^M = \min_{p,l}\left(\frac{C_q(p,l)}{\mathcal{P}_{qM}(p,l)}\right),$$

where

$$C_q(p,l) = \max\left(\sqrt{q\binom{k+l}{p}}, \frac{q\binom{k+l}{p}}{2^l}\right), q \leq \binom{k+l}{p}$$

is the complexity of solving the DOOM problem using Dumer's algorithm and

$$\mathcal{P}_{qM}(p,l) = 1 - \left(1 - \frac{\binom{n-k-l}{w-p}\binom{k+l}{p}}{\binom{n}{w}}\right)^{qM}$$

is the success probability. This work factor is the reference for choosing the parameters of the signature scheme [25].

Although advanced algorithms for information set decoding can be adapted to DOOM to reduce complexity, this has not yet been conducted. The proposed signature scheme is designed to use codes with a high-dimensional hull. Hence, the attacker can exploit this. However, to our knowledge, there is no algorithm for information set decoding or DOOM that considers this.

With our computation, the DOOM work factor value of `Enh-pqsigRM-613` is 2^{259}.

Security against key substitution attacks : In a key substitution attack, the adversary attempts to find a valid key that is different from the correct key and can be used for signature verification. If the adversary knows the secret key and the public key corresponding to a message–signature pair, we have a weak-key substitution attack, whereas if the adversary knows only the public key, we have a strong-key substitution attack. Both polynomial-time weak- and strong-key substitution attacks on the CFS signature scheme were proposed in [26]. A modification of the CFS scheme that resists such attacks was also proposed in [26]. In this modification, the syndrome $\mathbf{s}$ is generated by hashing the message, counter, and public key, rather than hashing only the message and counter. It has been demonstrated that this modified CFS signature scheme is secure against key substitution attacks [27]. In the Enhanced pqsigRM, the syndrome is given as $\mathbf{s} = h(M|i)$, and thus it is also secure against key substitution attacks [25].

EUF-CMA security : Here, we prove the EUF-CMA security of the modified pqsigRM. The methods presented below are adapted from the EUF-CMA security proof of SURF and Wave [10, 11]. It should be noted that although a key attack for SURF is presented in [10], its proof technique is valid and generally applicable. The proof is essentially the same except for the code used for the key and the decoding algorithm for signing.

1) Basic techniques for EUF-CMA security proof : EUF-CMA is a widely used attack model against signature schemes. In the security reduction task, EUF-CMA is viewed as a game played between an adversary and a challenger. The public key PK, hash oracle $\mathcal{H}$, and signing oracle Σ are given to a $(t, q_{\mathcal{H}}, q_{\Sigma}, \epsilon)$-adversary $\mathcal{A}$, where $\mathcal{A}$ can query at most $q_{\mathcal{H}}$ hash values and q_{Σ} signatures for inputs of its own choice. Within a maximum computation time t, $\mathcal{A}$ attempts to find a valid message–signature pair $(\mathbf{m}^*, \sigma^*)$. $\mathcal{A}$ wins the game if Verifying$(\mathbf{m}^*, \sigma^*, PK) = 1$ and σ^* has not been provided by Σ; otherwise, the challenger wins the game. The winning probability of the $(t, q_{\mathcal{H}}, q_{\Sigma}, \epsilon)$-adversary is at least ϵ.

Definition 1. (EUF-CMA security)
Let $\mathcal{S}$ be a signature scheme. We define the EUF-CMA *success probability against $\mathcal{S}$ as*

$$Succ_{\mathcal{S}}^{\mathrm{EUF-CMA}}(t, q_{\mathcal{H}}, q_{\Sigma}) := \max(\epsilon | \exists (t, q_{\mathcal{H}}, q_{\Sigma}, \epsilon)\text{-}adversary).$$

The signature scheme $\mathcal{S}$ is called $(t, q_{\mathcal{H}}, q_{\Sigma})$-secure in EUF-CMA *if the above success probability is a negligible function of the security parameter λ.*

We use the statistical and computational distance as basic metrics.

Definition 2. (Statistical distance)
The statistical distance between two discrete probability distributions $\mathcal{D}^0$ and $\mathcal{D}^1$ over the same space $\mathcal{E}$ is defined as

$$\rho(\mathcal{D}^0, \mathcal{D}^1) := \frac{1}{2} \sum_{x \in \mathcal{E}} |\mathcal{D}^0(x) - \mathcal{D}^1(x)|.$$

Proposition 1. [10] *Let $(\mathcal{D}_1^0, \ldots, \mathcal{D}_n^0)$ and $(\mathcal{D}_1^1, \ldots, \mathcal{D}_n^1)$ be two n-tuples of discrete probability distributions over the same space. For all $n \geq 0$, we have*

$$\rho(\mathcal{D}_1^0 \otimes \cdots \otimes \mathcal{D}_n^0, \mathcal{D}_1^1 \otimes \cdots \otimes \mathcal{D}_n^1) \leq \sum_{i=1}^{n} \rho(\mathcal{D}_i^0, \mathcal{D}_i^1).$$

Definition 3. (Computational distance and indistinguishability)
The computational distance between two distributions $\mathcal{D}^0$ and $\mathcal{D}^1$ in time t is

$$\rho_c(\mathcal{D}^0, \mathcal{D}^1) := \frac{1}{2} \max_{|\mathcal{A}| \leq t} \left(Adv^{\mathcal{D}^0, \mathcal{D}^1}(\mathcal{A}) \right),$$

where $|\mathcal{A}|$ denotes the running time of $\mathcal{A}$, and $Adv^{\mathcal{D}^0,\mathcal{D}^1}$ is the advantage of distinguisher $\mathcal{A}$, which returns $b \in \{0,1\}$ against $\mathcal{D}^0$ and $\mathcal{D}^1$:

$$Adv^{\mathcal{D}^0,\mathcal{D}^1} \quad := \quad \mathbb{P}_{\xi\sim\mathcal{D}^0}(\mathcal{A}(\xi) \textit{ outputs } 1) \quad - \quad \mathbb{P}_{\xi\sim\mathcal{D}^1}(\mathcal{A}(\xi) \textit{ outputs } 1).$$

The EUF-CMA security of the *Enhanced pqsigRM* is reduced to the *modified RM code distinguishing problem* and *DOOM with a high-dimensional hull.*

Problem 2. (Modified RM code distinguishing problem)

Instance: A code $\mathcal{C}$ with a high-dimensional hull.
Output: A bit $b \in \{0,1\}$, where $b=1$ if $\mathcal{C}$ is a permutation of the modified RM code; otherwise, $b=0$.

Problem 3. (DOOM with a high-dimensional hull)

Instance: A parity check matrix $\mathbf{H}' \in \mathbb{F}_2^{(n-k)\times n}$ of an (n,k) code with a high-dimensional hull, syndromes $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_q \in \mathbb{F}_n^{(n-k)}$, and an integer w.
Output: $(\mathbf{e}, i) \in \mathbb{F}_2^n \times [1,q]$ such that $\mathrm{wt}(\mathbf{e}) \leq w$ and $\mathbf{He}^T = \mathbf{s}_i^T$.

Definition 4. (One-wayness of DOOM with a high-dimensional hull)
We define the success of an algorithm $\mathcal{A}$ against DOOM with a high-dimensional hull and parameters n,k,q,w as

$$Succ^{n,k,q,w}(\mathcal{A}) \quad = \quad \mathbb{P}(\mathcal{A}(\mathbf{H}, \mathbf{s}_1, \ldots, \mathbf{s}_q) \textit{ is a solution of Problem 3}),$$

where $\mathbf{H}$ is chosen uniformly from the parity check matrix of (n,k) codes with a high-dimensional hull, $\mathbf{s}_i$ is chosen uniformly in $\mathbb{F}_2^{n-k}$, and the probability is taken over these choices and the internal coin of algorithm $\mathcal{A}$. The computational success of breaking DOOM with a high-dimensional hull in time t is defined by

$$Succ^{n,k,q,w}_{DOOMHull}(t) = \max_{|\mathcal{A}|\leq t} \left(Succ^{n,k,q,w}(\mathcal{A})\right).$$

We assume here that the probability is negligible (as a function of λ) for the parameters given in Table 3.

It is worth noting that there are sufficiently many codes with high-dimensional hulls for the parameters given in Tables 5 and 3 [28].

Proof of EUF-CMA security Let $\mathcal{S}_{pqsigRM}$ denote the proposed modified pqsigRM. The following definitions as well as the theorem and its proof are adopted from those in [10, 11].

Definition 5. (Challenger procedures in the EUF-CMA game)
The challenger procedures in the EUF-CMA game corresponding to $\mathcal{S}_{pqsigRM}$ are defined as follows:

`proc Init`(λ)	`proc Hash`$(\mathbf{m}, \mathbf{i})$
$(PK, SK) \leftarrow$ `Gen`(1^λ) $\mathbf{H}' \leftarrow PK$ $(\mathbf{H}, \mathbf{S}, \mathbf{Q}) \leftarrow SK$ `return` $\mathbf{H}'$	`return` $h(\mathbf{m}, \mathbf{i})$
`proc Sign`$(\mathbf{m})$	`proc Finalize`$(\mathbf{m}, \mathbf{e}, \mathbf{i})$
$\mathbf{i} \hookleftarrow \{0,1\}^{\lambda_0}$ $\mathbf{s} \leftarrow$ `Hash`$(\mathbf{m}, \mathbf{i})$ $\mathbf{e} \leftarrow \textsc{Decode}(\mathbf{S}^{-1}\mathbf{s}^T; \mathbf{H})$ `return` $(\mathbf{eQ}, \mathbf{i})$	$\mathbf{s} \leftarrow$ `Hash`$(\mathbf{m}, \mathbf{i})$ `return` $\mathbf{H}'\mathbf{e}^T = \mathbf{S}^T \wedge wt(\mathbf{e}) = w$

We note that the procedures in Definition 5 simplify Algorithm 4. We can now modify the security reduction in [10, 11] and prove the EUF-CMA security of the modified pqsigRM as follows.

Theorem 1. (Security reduction)
Let $Succ^{\mathrm{EUF-CMA}}_{\mathcal{S}_{pqsigRM}}(t, q_{\mathcal{H}}, q_{\Sigma})$ be the success probability of the EUF-CMA game corresponding to $\mathcal{S}_{pqsigRM}$ for time t when the number of queries to the hash oracle (resp. signing oracle) is $q_{\mathcal{H}}$ (resp. q_{Σ}). Then, in the random oracle model, we have for all t

$$Succ^{\mathrm{EUF-CMA}}_{\mathcal{S}_{pqsigRM}}(t, q_{\mathcal{H}}, q_{\Sigma}) \leq 2Succ^{n,k,q,w}_{DOOM\,Hull}(t_c) + q_{\mathcal{H}}\mathbb{E}_{\mathbf{H}'}\left(\rho(\mathcal{D}^{\mathbf{H}'}_w, \mathcal{U}_s)\right) + q_{\Sigma}\rho(\mathcal{D}_w, \mathcal{U}_w) + \rho_c(\mathcal{D}_{pub}, \mathcal{D}_{rand})(t_c) + \frac{1}{2^\lambda},$$

where $t_c = t + O(q_{\mathcal{H}} \cdot n^2)$, $\mathcal{D}^{\mathbf{H}'}_w$ is the distribution of the syndromes $\mathbf{H}'\mathbf{e}^T$ when $\mathbf{e}$ is drawn uniformly from the binary vectors of weight w, $\mathcal{U}_s$ is the uniform distribution over $\mathbb{F}_2^{n-k}$, $\mathcal{D}_w$ is the distribution of the decoding result of Algorithm 3, $\mathcal{U}_w$ is the uniform distribution over the binary vectors of weight w, $\mathcal{D}_{rand}$ is the uniform distribution over the random codes with high-dimensional hulls, and $\mathcal{D}_{pub}$ is the uniform distribution over the public keys of modified pqsigRM.

Proof. *Let $\mathcal{A}$ be a $(t, q_{\mathcal{H}}, q_{\Sigma}, \epsilon)$-adversary against $\mathcal{S}_{pqsigRM}$, and let $(\mathbf{H}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{q_{\mathcal{H}}})$ be a random instance of DOOM with a high-dimensional hull for the parameters $n, k, q_{\mathcal{H}}$, and w. We stress that $\mathbf{s}_1, \ldots, \mathbf{s}_{q_{\mathcal{H}}}$ are random independent vectors of $\mathbb{F}_2^{n-k}$. Let $\mathbb{P}(S_i)$ denote the probability that $\mathcal{A}$ wins Game i.*

Game 0 *is the EUF-CMA game for $\mathcal{S}_{pqsigRM}$.*

Game 1 *is the same as Game 0 except for the following failure event F: There is a collision in a signature query. From the difference lemma in [29], we have*

$$\mathbb{P}(S_1) \leq \mathbb{P}(S_0) + \mathbb{P}(F). \tag{1}$$

The following lemma is from [11].

Lemma 1. *For $\lambda_0 = \lambda + 2\log_2(q_{\mathcal{H}})$, we have $\mathbb{P}(F) \leq \frac{1}{\lambda}$.*

```
proc Hash($\mathbf{m}, \mathbf{i}$)
if $\mathbf{i} \in L_{\mathbf{m}}$
  $\mathbf{e}_{\mathbf{m},\mathbf{i}} \hookleftarrow S_w$
  return $\mathbf{H}'\mathbf{e}_{\mathbf{m},\mathbf{i}}^T$
else
  $j \leftarrow j + 1$
  return $\mathbf{s}_j$
```

```
proc Sign($\mathbf{m}$)
$\mathbf{i} \leftarrow L_{\mathbf{m}}$.next()
s $\leftarrow$ Hash($\mathbf{m}, \mathbf{i}$)
$\mathbf{e} \leftarrow$ DECODE($\mathbf{S}^{-1}\mathbf{s}^T; \mathbf{H}$)
return ($\mathbf{eQ}, \mathbf{i}$)
```

Game 2 *is obtained from Game 1 by changing **Hash** and **Sign** as follows, where S_w denotes the set of vectors with Hamming weight w in $\mathbb{F}_2^n$:*

*Index j is initialized to 0 in the **Init** procedure. We introduce the list $L_{\mathbf{m}}$, which contains $q_{\mathcal{H}}$ random elements of $\mathbb{F}_2^{\lambda_0}$ for each message $\mathbf{m}$. The list is sufficiently large so that all queries are satisfied. The **Hash** procedure returns $\mathbf{H}'\mathbf{e}_{\mathbf{m},\mathbf{r}}^T$ if and only if $\mathbf{i} \in L_{\mathbf{m}}$; otherwise, it returns $\mathbf{s}_j$. The **Sign** process is unchanged unless $\mathbf{i} \in L_{\mathbf{m}}$.*

*The statistical distance between the syndromes generated by matrix $\mathbf{H}'$ and the uniform distribution over $\mathbb{F}_2^{n-k}$ is $\rho(\mathcal{D}_w^{\mathbf{H}'}, \mathcal{U}_s)$. This is the difference between **Hash** in Game 1 and Game 2 when $\mathbf{i} \in L_{\mathbf{m}}$. There are at most $q_{\mathcal{H}}$ such instances. Thus, by Proposition 1, it follows that*

$$\mathbb{P}(S_2) \leq \mathbb{P}(S_1) + q_{\mathcal{H}}\mathbb{E}_{\mathbf{H}'}\left(\rho(\mathcal{D}_w^{\mathbf{H}'}, \mathcal{U}_s)\right). \tag{2}$$

Game 3 *is obtained from Game 2 by replacing* DECODE *with $\mathbf{e}_{\mathbf{m},\mathbf{i}}$ in **Sign** procedure as follows: $\mathbf{e}$ is drawn according to the proposed decoding algorithm* DECODE *in Game 2, whereas it is now drawn according to the uniform distribution $\mathcal{U}_w$. By Proposition 1, we have*

Game 3

```
proc Sign($\mathbf{m}$)
$\mathbf{i} \leftarrow L_{\mathbf{m}}$.next()
s $\leftarrow$ Hash($\mathbf{m}, \mathbf{i}$)
$\mathbf{e} \leftarrow \mathbf{e}_{\mathbf{m},\mathbf{i}}$
return ($\mathbf{e}, \mathbf{i}$)
```

Game 5

```
proc Finalize($\mathbf{m}, \mathbf{e}, \mathbf{i}$)
s $\leftarrow$ Hash($\mathbf{m}, \mathbf{i}$)
$b \leftarrow \mathbf{H}'\mathbf{e}^T = \mathbf{S}^T \wedge wt(\mathbf{e}) = w$
return $b \wedge (\mathbf{i} \notin L_{\mathbf{m}})$
```

$$\mathbb{P}(S_3) \leq \mathbb{P}(S_2) + q_{\Sigma}\rho(\mathcal{D}_w, \mathcal{U}_w). \tag{3}$$

Game 4 *is the game in which $\mathbf{H}'$ is replaced with $\mathbf{H}_0$. This implies that the adversary is forced to construct a solution for DOOM with a high-dimensional hull. Here, if a difference between Game 3 and Game 4 is detected, then this yields a distinguisher between $\mathcal{D}_{pub}$ and $\mathcal{D}_{rand}$. According to [10], the cost to call **Hash** does not exceed $O(n^2)$, and thus the running time of the challenger is $t_c = t + O(q_{\mathcal{H}} \cdot n^2)$. Therefore, we have*

$$\mathbb{P}(S_4) \leq \mathbb{P}(S_3) + \rho_c(\mathcal{D}_{pub}, \mathcal{D}_{rand})(t_c). \tag{4}$$

Game 5 *is modified in* ***`Finalize`****. The success of Game 5 implies* $\mathbf{i} \notin L_{\mathbf{m}}$ *and the success of Game 4. A valid forgery* $\mathbf{m}^*$ *has never been queried by* ***`Sign`****, and the adversary has never accessed* $L_{\mathbf{m}^*}$*. As there are* q_Σ *signing queries, we have*

$$\mathbb{P}(S_5) = (1 - 2^{\lambda_0})^{q_\Sigma}\mathbb{P}(S_4).$$

Moreover, $(1 - 2^{\lambda_0})^{q_\Sigma} \geq \frac{1}{2}$ *because we assumed* $\lambda_0 = \lambda + 2\log_2(q_\Sigma)$*. Thus, this can be simplified to*

$$\mathbb{P}(S_5) \geq \frac{1}{2}\mathbb{P}(S_4). \tag{5}$$

$\mathbb{P}(S_5)$ *is the probability that* $\mathcal{A}$ *returns a solution for DOOM with a high-dimensional hull, which yields*

$$\mathbb{P}(S_4) \leq 2Succ^{n,k,q,w}_{DOOMHull}(t_c). \tag{6}$$

Combining (1)–(6) concludes the proof.

Complexity of finding minimum weight codewords : Using information set decoding, the probability of successful decoding of a weight-w-error vector is as follows.

$$Prob(Dec) = \frac{\binom{n-k}{w}}{\binom{n}{w}} = \frac{(n-k)(n-k-1)\cdots(n-k-w+1)}{n(n-1)\cdots(n-w+1)} \approx (\frac{n-k}{n})^w \tag{7}$$

This probability works the same as finding the minimum weight codewords problem when syndrome equals 0. Thus, we can get the same equation with (1). In other words, the complexity of finding minimum weight codewords is the inverse of (1) substituting w to d_{min} as

$$Complexity = (\frac{n}{n-k})^{d_{min}}. \tag{8}$$

We compute this with `Enh-pqsigRM-613` and the result of complexity is 2^{128}. That means, it still satisfies 128-bit security.

6.3 Indistinguishability of Codes and Signature in the Proposed Scheme

Modifications of public codes : Cryptanalysis using hulls is widely used in code-based cryptography. However, this is valid if the hull has a specific structure that allows information leakage about the secret key. Therefore, using only the fact that the dimension of the hull is large, it is difficult to distinguish whether the code is public or random code with a high-dimensional hull. The EUF-CMA security proof requires the indistinguishability between public and random codes. We will discuss the design methodology and how these modifications can ensure indistinguishability. Considering the key recovery attack in [10], a $(U, U+V)$-code used in code-based crypto-algorithms should have a high-dimensional hull for security. Even though the public code of the proposed signature scheme is not a $(U, U+V)$-code, it should contain a $(U, U+V)$ subcode for efficient decoding. The attack on SURF in [10] uses the fact that for any $(U, U+V)$-code, the hull of the public code is highly probable to have a $(\mathbf{u}|\mathbf{u})$ structure when $U^{\perp} \cap V = \{\mathbf{0}\}$, $dim(U) \geq dim(V)$. This $(\mathbf{u}|\mathbf{u})$ reveals information about the secret permutation Q and enables the attacker to locate the U and $U+V$ codes. To avoid this, we should maintain the high dimension of $U^{\perp} \cap V$, implying that the public code should have a high-dimensional hull. Hence, we define DOOM with a high-dimensional hull and assume that the public code of Enhanced pqsigRM is indistinguishable from a random code with a hull of the same dimension as that of the public code, rather than any random linear code.

Moreover, k_{app} random rows are appended to the generator matrix, and 2^r rows of the generator matrix, that is the repeated $\mathrm{RM}_{(r,r)}$, are replaced by k_{rep} random rows; furthermore, a codeword from the dual code is appended to the generator matrix. These modifications are equivalent to increasing the dimension of the code itself, the hull, and the dual of the code, respectively, by appending random codewords. Moreover, by adding random codewords, the code is no longer a $(U, U+V)$-code, and thus distinguishing attacks are more difficult to perform. We now explain the rationale for the aforementioned modifications, which are applied in addition to partial permutation.

1) k_{app} random rows are appended to the generator matrix : The Hamming weights of a random code are distributed. However, the partially permuted RM code has only codewords with even Hamming weight. This is because the Hamming weights of codewords of $\mathrm{RM}_{(r,m)}$ are even numbers, and partial permutations do not affect parity.

By appending a random row with an odd hamming weight to the generator matrix, the Hamming weights of the public code become distributed binomially. The problem is that if only one row with an odd Hamming weight is appended, it can easily be extracted. This can be resolved by appending more than one codeword. Hence, we append k_{app} random rows such that at least one has an odd Hamming weight. By the nature of the decoding process, it is still possible to decode the resulting code.

2) Appending a random codeword of the dual code to the generator matrix : The Hamming weights of the codewords in the hull of the partially permuted RM code are only multiples of four. However, the Hamming weight of the codewords in the hull of a random code may be an arbitrary even number, not only a multiple of four. As in the previous modification, a random codeword is appended to the hull. Thereby, we force the codewords of the hull of the public code to have arbitrary even Hamming weights. As a randomly appended row to the generator matrix is unlikely to be appended to its hull, appending a codeword to the hull is more complicated. The following is the process for appending a random codeword to the hull.

Let $hull(\mathcal{C})$ be the hull of a code $\mathcal{C}$. We define $\mathcal{C}'$ and $\mathcal{C}''$ by $\mathcal{C} = hull(\mathcal{C}) + \mathcal{C}'$ and $\mathcal{C}^{\perp} = hull(\mathcal{C}) + \mathcal{C}''$, where $hull(\mathcal{C})$, $\mathcal{C}'$, and $\mathcal{C}''$ are linearly independent. We can then generate a code with a hull with dimension $dim(hull(\mathcal{C})) + 1$ by the following procedure:

i) Find a codeword $\mathbf{c}_{dual} \in \mathcal{C}''$ such that $\mathbf{c}_{dual} \cdot \mathbf{c}_{dual} = 0$. This is easy because a codeword with even Hamming weight satisfies it.
ii) Let $\mathcal{C}_{inc} = \mathcal{C} + \{\mathbf{c}_{dual}\} = (hull(\mathcal{C}) + \{\mathbf{c}_{dual}\}) + \mathcal{C}'$.
iii) As $\mathbf{c}_{dual} \cdot (hull(\mathcal{C}) + \{\mathbf{c}_{dual}\}) = \{0\}$ and $\mathbf{c}_{dual} \cdot \mathcal{C}' = \{0\}$, we have $\mathbf{c}_{dual} \in \mathcal{C}_{inc}^{\perp}$, where for a vector x and a set of vectors A, $x \cdot A$ is the set of all inner products of x and elements of A.
iv) It can be seen that $\mathcal{C}_{inc} \cap \mathcal{C}_{inc}^{\perp} = (hull(\mathcal{C}) + \{\mathbf{c}_{dual}\})$. Hence, $\mathcal{C}_{inc}$ is a code that has a hull of which dimension is $dim(hull(\mathcal{C})) + 1$.

If the Hamming weights of the codewords of the hull are only multiples of 4, then another c_{dual} is selected, and the above process is repeated.

3) Repeated RM$_{(r,r)}$ is replaced with random $(2^r, k_{rep})$ codes : We note that by replacing repeated RM$_{(r,r)}$ by random $(2^r, k_{rep})$ codes, the dimension of the code is reduced by $2^r - k_{rep}$; this is equivalent to appending $2^r - k_{rep}$ rows to the parity check matrix. The codewords of the dual code of the partially permuted RM code have only codewords of even Hamming weight owing to a subcode of the partially permuted RM code. This can be resolved by replacing this subcode with another random code such that its MD decoder exists. The partially permuted RM code includes (RM$_{(r,r)}| \ldots |$RM$_{(r,r)}$), and the dual code of this has only codewords of even Hamming weight by the proposition below. It is easy to verify that the dual code of the partially permuted RM code is a subset of the dual code of (RM$_{(r,r)}| \ldots |$RM$_{(r,r)}$). That is, (RM$_{(r,r)}| \ldots |$RM$_{(r,r)}$) causes the dual code of the partially permuted RM code to have only codewords of even Hamming weight. By replacing the repeated RM$_{(r,r)}$ with a random code such that its dual code has codewords of odd Hamming weight, we can force the dual of the public code to have codewords with an odd hamming weight.

Clearly, the dual code of RM$_{(r,r)}$ is $\{\mathbf{0}\}$. We replace RM$_{(r,r)}$ with a random $(2^r, k_{rep})$ code. We note that the dual code of this $(2^r, k_{rep})$ code must have codewords with an odd hamming weight. The generator matrix is modified in this manner, rather than by appending rows to the parity check matrix, to ensure that the entire code is decodable.

Public codes indistinguishability : In the EUF-CMA security proof, the modified RM code distinguishing problem should be hard. As it is challenging to find the computational distance between public and random codes, in this section, we study the randomness of the public code and consider possible attacks.

1) Public code is not a $(U, U + V)$-code : After random rows have been appended to the generator matrix of a $(U, U + V)$-code, the resulting code is unlikely to be a $(U, U+V)$-code. Considering the following proposition, it can be seen that with probability $O(2^{k_U - n/2})$, a $(U, U+V)$-code remains a $(U, U+V)$-code after a row has been appended to its generator matrix.

Proposition 2. *Let $\mathcal{C}$ be a $(U, U+V)$-code. Then, for all codewords $(\mathbf{c}'|\mathbf{c}'') \in \mathcal{C}, (\mathbf{0}|\mathbf{c}' - \mathbf{c}'') \in \mathcal{C}$.*

It is expected that attacking the modified RM code is difficult because the appended codewords change the algebraic structure of the code (i.e., the $(U, U+V)$ structure), there is considerable randomness, and there is currently no recovery algorithm.

2) Distinguishing using hull : When a random row is appended to the generator matrix, it is unlikely to be included in the hull. To achieve this, the appended row should be a codeword of the dual code, and its square should be zero. Hence, we append a codeword from the dual code to the generator matrix.

The appended row can be omitted when the attacker collects several independent codewords with Hamming weight 4 from the hull. However, for any random code with a high-dimensional hull, the same process can be applied, and finally, there only remain codewords of which the Hamming weight is a multiple of 4. Hence, this is not a valid distinguishing attack.

The hull of a random $(U, U+V)$-code is $\{\mathbf{0}\}$ when $k_U < k_V$ and is highly probable to have codewords of $(\mathbf{u}|\mathbf{u})$ form when $k_U \geq k_V$. However, the hull of an RM code is also an RM code, and in our case, the partial permutation randomizes its hull and retains its large dimension. The hull is neither a subcode of the RM code nor a $(U, U+V)$-code. Moreover, most of the hull depends on the secret partial permutations σ_p^1 and σ_p^2.

Signature leaks : In the EUF-CMA security proof, the indistinguishability between public and random codes should be guaranteed. If this is true, then the signature does not leak information. In several signature schemes, such as Durandal, SURF, and Wave, this is achieved and proved. In SURF and Wave, the rejection sampling method is applied to render the public code's indistinguishability.

To apply rejection sampling, the distribution of the decoding output should be known. In SURF and Wave, a simple and efficient decoding algorithm is used, and thus it is easy to find the distribution of the decoding output. However, in our case, the decoding output exhibits a high degree of randomness, and the structure of the decoder is complex. Therefore, it is difficult to analyze the distribution of the decoding output. Instead, we conduct a proof-of-concept implementation of the Enhanced pqsigRM using SageMath. Then, we perform statistical randomness tests under NIST SP 800-22 [30] on the decoding output, and we compare the results with random errors in $\mathbb{F}_2^n$ with Hamming weight w. No significant difference is observed. However, it should be noted that the success of a statistical randomness test does not imply indistinguishability. Thus, the indistinguishability of the signature should be rigorously studied in future work.

7 Conclusion

We introduced a new signature scheme, called Enhanced pqsigRM, based on modified RM codes with partial permutation as well as row appending and replacement in the generator matrix. For any given syndrome, an error vector with a small Hamming weight can be obtained. Moreover, the decoding method achieves indistinguishability to some degree because it is collision-resistant. The proposed signature scheme resists all known attacks against cryptosystems based on the original RM codes. The partially permuted RM code improves the signature success condition in previous signature schemes such as CFS and can improve signing time and key size.

We further modified the RM code using row appending/replacement. The resulting code is expected to be indistinguishable from random codes with the same hull dimension; moreover, the decoding of the partially permuted RM code is maintained. Assuming indistinguishability and the hardness of DOOM with a high-dimensional hull, we could achieve the EUF-CMA security of the proposed signature scheme.

Moreover, the *Enhanced pqsigRM* signature scheme has advantages in signature size. It has a relatively small signature size compared with the other digital signature NIST PQC finalist algorithms and code-based signatures. Also, it has a very short verification time for 128-bit security. The limitation of this scheme is the relatively large public key size. Since the code in Enhanced pqsigRM does not have a structure such as quasi-cyclic, the key size of the public key is $(n-k) \times k$.

For 128 bits of classical security, the signature size of the proposed signature scheme is 1032 bytes, which corresponds to 0.42 times that of Crystals-Dilithium, and the number of median verification cycles is 235,656, which corresponds to about 0.72 times that of Crystals-Dilithium. We expect the verification cycles to be shortened because the signing process is not complicated. We are working on it to reduce more.

8 Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00400, Development of Highly Efficient PQC Security and Performance Verification for Constrained Devices).

We would like to thank Dr. Ray Perlner of NIST for his valuable comments on our previous submission.

References

1. N. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *Proc. Asiacrypt*, vol. 2248, 2001, pp. 157–174.
2. J.-C. Faugere, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high-rate McEliece cryptosystems," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6830–6844, Oct 2013.
3. F. Hemmati, "Closest coset decoding of $u|u + v|$ codes," *IEEE J. Sel. Areas Commun.*, vol. 7, pp. 982–988, Aug. 1989.
4. I. Dumer, "Recursive decoding and its performance for low-rate Reed–Muller codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 811–823, May 2004.
5. L. Minder and A. Shokrollahi, "Cryptanalysis of the Sidelnikov cryptosystem," in *Proc EUROCRYPT* 2007, LNCS, vol. 4515, 2007, pp. 347–360.
6. I. V. Chizhov and M. A. Borodin, "The failure of McEliece PKC based on Reed–Muller codes," *IACR Cryptology ePrint Archive*, Report 2013/287 (2013).
7. W. Lee, Y. S. Kim, Y. W. Lee, and J. S. No, "Post quantum signature scheme based on modified Reed–Muller code pqsigRM," in *First Round Submission to the NIST Postquantum Cryptography Call*, Nov. 2017. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Round-1-Submissions.
8. I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Transactions of the IRE Professional Group on Information Theory,* vol. 4, no. 4, pp. 38-49, Sep. 1954.
9. D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *Transactions of the IRE Professional Group on Electronic Computers,* no. 3, pp. 6-12, Sep. 1954.
10. T. Debris-Alazard, N. Sendrier, and J.-P. Tillich. "SURF: A new code-based signature scheme," *arXiv preprint*, arXiv:1706.08065, 2017.
11. T. Debris-Alazard, N. Sendrier, and J.-P. Tillich, "Wave: A new family of trapdoor one-way preimage sampleable functions based on codes," in *International Conference on the Theory and Application of Cryptology and Information Security,* 2019: Springer, pp. 21-51.
12. N. Sendrier, "Decoding one out of many," *International Workshop on Post-Quantum Cryptography*, Springer, Berlin, Heidelberg, 2011.
13. Knuth, *Seminumerical Algorithms.* The Art of Computer Programming. 2 (3rd ed.). Boston: Addison–Wesley. pp. 145-–146. ISBN 0-201-89684-2. OCLC 38207978.
14. V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehle, and S. Bai, "Crystals-Dilithium," *NIST PQC Standardization Process Round 3 Submission*, 2021.
15. T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon," *NIST PQC Standardization Process Round 3 Submission*, 2021.
16. A. Hulsing, D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, P. Kampanakis, S. Kolbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, J.-P. Aumasson, B. Westerbaan, and W. Beullens, "Sphincs+," *NIST PQC Standardization Process Round 3 Submission*, 2021.
17. M. Finiasz, "Parallel-CFS," in *Sel. Areas in Cryptography,* Waterloo, Ontario, Canada, 2010, pp. 159–170.

18. N. Aragon, O. Blazy, P. Gaborit, A. Hauteville, and G. Zémor, "Durandal: a rank metric based signature scheme," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2019: Springer, pp. 728-758.
19. M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. Maurich, R. Misoczki, R. Niederhagen, K. G. Patterson, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, C. J. Tjhai, M. Tomlinson, and W. Wang, "Classic McEliece," *NIST PQC Standardization Process Round 3 Submission*, 2021.
20. A. Otmani and H. T. Kalachi, "Square code attack on a modified Sidelnikov cryptosystem," in *Proc. C2SI*, 2015, pp. 173–183.
21. J. Stern, "A method for finding codewords of small weight," *Coding Theory and Applications*, vol. 388, pp. 106-133, 1989.
22. I. Dumer, "On minimum distance decoding of linear codes," in *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory,* 1991, pp. 50-52.
23. A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques,* 2012: Springer, pp. 520-536.
24. A. May and I. Ozerov, "On computing nearest neighbors with applications to decoding of binary linear codes," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques,* 2015: Springer, pp. 203-228.
25. Y. Lee, W. Lee, Y.-S. Kim, and J.-S. No, "A modified pqsigRM: RM code-based signature scheme," *Cryptology ePrint Archive, Report 2019/678*, 2019.
26. B. Dou, C.-H. Chen, and H. Zhang, "Key substitution attacks on the CFS signature," *IEICE Transactions on Fundamentals of Electronics,* Communications and Computer Sciences, vol. 95, no. 1, pp. 414-416, Jan. 2012.
27. K. Morozov, P. S. Roy, R. Steinwandt, and R. Xu, "On the security of the Courtois-Finiasz-Sendrier signature," *Open Mathematics,* vol. 16, no. 1, pp. 161-167, 2018.
28. N. Sendrier, "On the dimension of the hull," *SIAM Journal of Discr. Math.,* vol. 10, no. 2, pp. 282-293, May 1997.
29. V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," *IACR Cryptology ePrint Archive,* vol. 2004, p. 332, 2004.
30. I. Lawrence et al., "SP 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Tech. NIST Rep.