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

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

[1] Seoul National University, South Korea
jsno@snu.ac.kr, jgjo114@ccl.snu.ac.kr
[2] Samsung Electronics, South Korea
yw0803.lee@samsung.com, bravokoo@gmail.com
[3] Chosun University, South Korea
iamyskim@chosun.ac.kr

**Abstract.** We present a novel code-based digital signature scheme, called enhanced pqsigRM for post-quantum cryptography (PQC). This scheme is based on a modified Reed–Muller (RM) code, which reduces the signature size and verification time compared with existing code-based signature schemes. In fact, it strengthens pqsigRM submitted to NIST for post-quantum cryptography standardization. The proposed scheme has the advantage of the short signature size and fast verification and uses public codes that are more difficult to distinguish from random codes. We use $(U, U + V)$-codes with the high-dimensional hull to overcome the disadvantages of code-based schemes. The proposed decoder samples from coset elements with small Hamming weight for any given syndrome and efficiently finds such an element. Using a modified RM code, the proposed signature scheme resists various known attacks on RM-code-based cryptography. It has advantages on signature size, verification time, and proven security. For 128 bits of classical security, the signature size of the proposed signature scheme is 512 bytes, which corresponds to 1/4.7 of that of CRYSTALS-DILITHIUM, and the number of median verification cycles is 759,248, which corresponds to the twice of that of CRYSTALS-DILITHIUM.

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

## 1 Introduction

A new digital signature scheme based on a modified Reed-Muller (RM) code is proposed. This signature scheme improves the Goppa code-based signature scheme developed by Courtois, Finiasz, and Sendrier (CFS) [1].

* This work is submitted to 'Korean Post-Quantum Cryptography Competition' (www.kpqc.or.kr).

The 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, but we need to significantly increase the key size itself for higher security. Also, with small $t$, the rate of Goppa code is high. The parity check matrix of high rate Goppa code can be distinguished from a random matrix and thus the CFS signature scheme is insecure under the EUF-CMA [2].

In this paper, we replace the Goppa code with the RM code in the CFS signature scheme. RM code 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 code with RM code in the CFS signature scheme results in vulnerability to several attacks. The RM code-based McEliece cryptosystem is insecure due to 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. It is also proved that the proposed enhanced pqsigRM is EUF-CMA secure.

We propose a new code-based signature scheme by using a modified RM code, called enhanced pqsigRM. We first partially permute the original RM code and proceed three more modifications, which are replacing some parts of the code, appending random rows, and padding a dual code's codeword. There are three enhanced pqsigRMs, that is, Enh-pqsigRM-412 constructed by RM(4,12) for 128 bit-security, and Enh-pqsigRM-413 constructed by RM(4,13) for 256 bit-security. The proposed signature scheme is an improvement of pqsigRM [7] submitted to NIST for PQC standardization, and it resolves the weaknesses of early versions of pqsigRM by modifying the public code. Moreover, we ensure the indistinguishability of the public code of the proposed signature scheme. Further, the proposed signature scheme 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 advantages on the small signature size, fast verifcation (for 128-bit security), and proven security. For 128 bits of classical security, the signature size of the proposed signature scheme is 512 bytes, which corresponds to 1/4.7 of that of CRYSTALS-DILITHIUM, and the number of median verification cycles is 759,248, which corresponds to the twice of 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. 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 modify 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 achieve the EUF-CMA security of the proposed signature scheme.

### 1.2 Advantages and limitations

Enhanced pqsigRM signature scheme has advantages on signature size and verification time. It has the smallest signature size compared with the other digital signature finalist algorithms. Also, it has a very short verification time for 128-bit security. Moreover, the security level is controllable by the parameter setting. 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 $(n-k) \times k$. Besides, it has relatively large number of verification cycles for 256-bit security. For 128 bits of classical security, the signature size of the proposed signature scheme is 512 bytes, which corresponds to 1/4.7 of that of CRYSTALS-DILITHIUM, and the number of median verification cycles is 759,248, which corresponds to the twice of that of CRYSTALS-DILITHIUM.

## 2 Preliminaries

### 2.1 CFS Signature Scheme

CFS signature scheme is an algorithm that applies the FDH methodology to the Niederreiter cryptosystem. The CFS signature scheme is based on Goppa codes, as McEliece cryptosystem. A summary of CFS signature scheme is given in Algorithm 1.

As described in Algorithm 1, the signing process iterates until a decodable syndrome is obtained. The probability that a given random syndrome can be decoded 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, the 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, $\approx$ keysize$^{t/2}$. Hence, the key size should be fairly large to meet a certain security level. In summary, the CFS signature scheme is insecure and inefficient owing to the use 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]. There are various definitions of RM codes, but we adopt a recursive definition here as recursive decoding is defined by using this structure. An RM code $\mathrm{RM}_{(r,m)}$ is a linear binary $(n =$

$2^m, k = \sum_{i=0}^{r} \binom{m}{i})$ code, 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 $\mathrm{RM}_{(0,m)} := \{(0,\dots,0),(1,\dots,1)\}$ with code length $2^m$ and $\mathrm{RM}_{(m,m)} := \mathbb{F}_2^{2^m}$. This is the well-known 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 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(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
        (y'|y'') ← y
        y^v = y' · y''
        v̂ ← RecursiveDecoding(y^v, r − 1, m − 1)
        y^u ← (y' + y'' · v̂)/2
        û ← RecursiveDecoding(y^u, r, m − 1)
        Output (û|û · v̂)
    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 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 $\sigma$, 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 $\sigma = (1, 3, 2, 0)$. 12‘

## 3.2 Specification of Enhanced pqsigRM

**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.

**Private Key and Public Key 1-1) Partial permutation of generator matrix of RM code:** Let $G(r, m)$ be a $k \times n$ generator matrix of the $(n, k)$ RM code, RM$(r, m)$. We know that the generator matrix of RM$(r, m)$ can be expressed as follows:

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

$$= \left[\begin{array}{cc:cc} G(r,m-2) & G(r,m-2) & G(r,m-2) & G(r,m-2) \\ \mathbf{0} & G(r-1,m-2) & \mathbf{0} & G(r-1,m-2) \\ \hdashline \mathbf{0} & \mathbf{0} & G(r-1,m-2) & G(r-1,m-2) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & G(r-2,m-2) \end{array}\right]. \tag{2}$$

The recursive decoding algorithm of RM code in [4] is possible from the recursive structure of the RM code.

For example, the decoding for the second half of the RM codeword can be done by decoding; i) the first half part of codeword by generator matrix $G(r, m-1)$, ii) the subcode generated by the lower right submatrix $G(r-1, m-1)$ in (1). Likewise, decoding for the first half of the second half of this code, i.e., $(c_{\frac{n}{2}+1}, \ldots, c_{\frac{3}{4}n})$ can be decoded by decoding the subcode generated by $G(r-1, m-2)$, which are submatrices of $G(r-1, m-1)$, i.e., two $G(r-1, m-2)$'s in the third row in (2).

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

In other words, we can say that $c = [u|u+v]$, for all $c \in \mathrm{RM}(r,m)$, where $u \in \mathrm{RM}(r,m-1)$ and $v \in \mathrm{RM}(r-1,m-1)$. Recursively, $\mathrm{RM}(r,m-1)$ and $\mathrm{RM}(r-1,m-1)$ are also $[u|u+v]$ structure codes until $r = 0$ or $r = m$. Here, if the code corresponding to $u$ or $v$ is replaced with a code other than the RM code, and the decoding of the replaced code is appropriately done, the entire code $c$ can be decoded [3].

The example of the generator matrix of such code is given in Fig. 1. We define $\sigma_p^1$ and $\sigma_p^2$ as two independent partial permutations which are random permutations of only $p$ columns out of $n/4$ columns. This structure allows to protect the codes from the attack using hull.

**1-2) Modification by replacing, appending, and padding :** When we make the recursive structure of RM codes repeatedly, we obtain $G_{RM(r,m)}$'s on the first $2^r$ rows. We replace these with $2^{m-r}$ number of same $(2^r, k_{rep})$-codes as in Figure 2. Additionally, the dual code of these $(2^r, k_{rep})$ random codes should include at least one non-zero codeword with an odd Hamming weight. This step makes the code to be robust from the attack using its dual code. Then, we append $k_{app}$ number of independent random codewords which has at least one non-zero codeword with an odd Hamming weight. By doing this, we can solve the distinguishing problem of RM codes. Lastly, we pad a codeword of the dual code of the whole code. This step prevents the leakage of the information of the hull. With these steps, we can obtain the modified RM codes, which are strong against all possible attacks or information leakage. Moreover, we can use a new decoding algorithm for this code as in Algorithm 3.

**1-3) Generation of $S, Q, \sigma_p^1, \sigma_p^2, E$, and $H_{\mathcal{M}}$:**

Let $S$ be an $(n-k)\times(n-k)$ non-singular matrix and $Q$ be an $n \times n$ permutation matrix. To generate $\sigma_p^1$ and $\sigma_p^2$, $p$ elements are chosen from the index set $\{0, 1, \ldots, n/4-1\}$, the chosen elements are randomly permuted while others are not. Let $G$ be a generator matrix of RM code constructed in a recursive structure as in (2). Now, $\sigma_p^1$ is used to permute the submatrices of $G$ corresponding to $\mathrm{RM}(r,m-2)$'s in the first row, and $\sigma_p^2$ is used to permute the generator matrix of $\mathrm{RM}(r-2,m-2)$ on the last row as in Fig. 1. Let $G[a:b,c:d]^{\sigma}$ be the

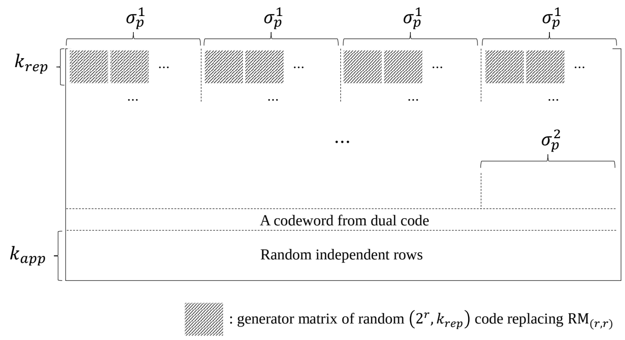


**Fig. 2.** Modified RM code's generator matrix $G_{\mathcal{M}}$ for the proposed signature scheme.

column permuted submatrix of $G$, where submatrix is composed of rows with indices $\{a, a+1, \ldots, b\}$ and columns with indices $\{c, c+1, \ldots, d\}$. Let $G_{\mathcal{M}}$ be the modified RM code's generator matrix of $\mathrm{RM}(r, m)$ in Fig. 2.

Thus, the dual matrix of the partially permuted generator matrix $G_{\mathcal{M}}$ becomes the parity check matrix, which is further modified into the row reduced echelon form (RREF), denoted by $H_{\mathcal{M}}$ to simplify signing. RREF of parity check matrix is always possible because parity check matrix has always full rank. Then, we compute $H' = SH_{\mathcal{M}}Q$, where the private keys are $S, Q, \sigma_p^1, and \sigma_p^2$. $Q$ is generated by a random shuffling algorithm using random numbers (such as Knuth's shuffling algorithm [11]) using random numbers generated by a random number generator based on AES-256 (shortly, RNG-AES-256). The two partial column permutations $\sigma_p^1$ and $\sigma_p^2$ are generated by i) permute $\{0, 1, \ldots, n/4-1\}$ by any shuffling algorithm ii) choose the first $p$ elements of the permuted sets, iii) elements other than these $p$ elements are placed in their original positions, and these $p$ elements are placed in the shuffled order in the remaining positions. We make this to be a systematic form as $H'_{sys} = (I \mid T)$. Then we can just use $T$ as a new public key which has a size of $(n-k) \times k$. This process reduces the public key size, considerably.

**Generation of Digital Signatures** 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 CFS signature scheme, we use hash function once, instead of twice with SHAKE-128, 256. Find the error vector $e$ such that $SH_{\mathcal{M}}Qe^T = s$. Let $e'^T = Qe^T$ and $s' = S^{-1}s$. Thus, $H_{\mathcal{M}}e'^T = s'$ holds. Then, perform decoding algorithm as in Algorithm 3 to find new error vector $e'$. If $\mathrm{wt}(e') \le w$, compute $e^T = Q^{-1}e'^T$, and the signature is then given as $(M, e, i)$, where $w$ is error weight parameter given in Section 3.

**Algorithm 3** Decoding for modified RM code

**function** $\text{DECODE}(\mathbf{s}; \mathbf{H})$
  $\mathbf{r} \leftarrow \text{PRANGE}(\mathbf{H}, \mathbf{s})$
  **while** True **do**
    $\mathbf{r} \leftarrow \mathbf{r}+$ random codeword
    $\mathbf{c} \leftarrow \text{MODDEC}(\mathbf{r}, r, M)$
    **if** $\text{wt}(\mathbf{r} + \mathbf{c}) \leq w$ **then**
      Output $\mathbf{r} + \mathbf{c}$
    **end if**
  **end while**
**end function**

**function** $\text{MODDEC}(\mathbf{y}, r, M)$
  $\mathbf{y} \leftarrow \mathbf{y}^{\sigma^{-1}}$
  **if** $r = 0$ **then**
    Output MD decoding on $\text{RM}(0, m)$
  **else if** $r = m$ **then**
    Output MD decoding on $\text{RM}(r, r)$
     or replaced $(2^r, k_{rep})$ code
  **else**
    $(\mathbf{y}'|\mathbf{y}'') \leftarrow \mathbf{y}$
    $\mathbf{y^v} = \mathbf{y}' \cdot \mathbf{y}''$
    $\hat{\mathbf{v}} \leftarrow \text{MODDEC}(\mathbf{y^v}, r-1, m-1)$
    $\mathbf{y^u} \leftarrow (\mathbf{y}' + \mathbf{y}'' \cdot \hat{\mathbf{v}})/2$
    $\hat{\mathbf{u}} \leftarrow \text{MODDEC}(\mathbf{y^u}, r, m-1)$
    $\mathbf{y} \leftarrow (\hat{\mathbf{u}}|\hat{\mathbf{u}} \cdot \hat{\mathbf{v}})$
  **end if**
  Output $\mathbf{y}^{\sigma}$
**end function**
*$\sigma$ is $\sigma_p^1$ or $\sigma_p^2$ for permuted block and identity, otherwise.

**Verification** Check $\text{wt}(e) \leq w$ and $H'_{sys}e^T = h(M|i)$. If TRUE, then return ACCEPT; else, return REJECT.

The key generation, signing, and verification processes of the enhanced pqsigRM are described in Algorithm 4.

### 3.3 Parameter sets

**Parameter Set Enh-pqsigRM-612** Uses RM code RM(6,12) with $w = 495$ and $p = 386$ (128-bit security).

**Parameter Set Enh-pqsigRM-613** Uses RM code RM(6,13) with $w = 1370$ and $p = 562$ (256-bit security).

The sizes of the public key and signature are given in Table 1. Comparing with the finalist schemes [20–22], our scheme has the smallest signature size.

**Algorithm 4** Signature scheme of *enhanced pqsigRM* [12]

**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
$\mathbf{S} \xleftarrow{\$} F_2^{(n-k)\times(n-k)}, \mathbf{Q} \xleftarrow{\$} F_2^{n\times n}$
$\mathbf{H}' \leftarrow \mathbf{S}\mathbf{H}_{\mathcal{M}}\mathbf{Q}$
$\mathbf{H}'_{\mathbf{sys}} = (\mathbf{I}|\mathbf{T})$ : systematic form of $\mathbf{H}'$
Public key: $\mathbf{T}$
Secret key: $\mathbf{H}_{\mathcal{M}}, \mathbf{S}, \mathbf{Q}$

**Signing** :

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

**Verification** :

If $wt(e) \leq w$ and $H'_{sys}e^T = h(M|i)$,
  return ACCEPT
Else, return REJECT

*$h$: hash function SHAKE-128/256
*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(Bytes) compared with the finalists

| Security | Enhanced pqsigRM | | CRYSTALS-DILITHIUM | | FALCON | | SPHINCS+ | |
|---|---|---|---|---|---|---|---|---|
| | Public key | Signature | Public key | Signature | Public key | Signature | Public key | Signature |
| 128 | 474,445 | 512 | 1,312 | 2,420 | 897 | 666 | 32 | 7,856 |
| 256 | 2,000,000 | 1,024 | 2,592 | 4,595 | 1,793 | 1,280 | 64 | 29,792 |

# 4 Performance analysis

## 4.1 Description of platform

The following measurements are collected using a desktop computer with CPU —i7-8700 CPU @ 3.20GHz— running at 3.40GHz. Turbo Boost is disabled. This machine has 32GB of RAM. Benchmarks have run on one core of the CPU. Since the signing algorithm is a probabilistic algorithm, the number of iteration at signing varies. The following result is the average of 100 experiments. For detailed descriptions of the success probability of the signing, see 2.B.1.

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

The following measurements are CPU cycler for running `Enh-pqsigRM-612`, `Enh-pqsigRM-613` at —i7-8700 CPU @ 3.20GHz—. The measurements compared with the finalists are given in Table 2. The data of the finalists are from the submission papers and these can be little bit different because they implementation conditions are different [20–22]. However, it is almost same with CRYSTALS-DILITHIUM. Comparing with CRYSTALS-DILITHIUM, we have about a twice of verification cycles for 128-bit security. However, we have about ten times larger cycles for 256-bit security. This implementation result is for the proof of concept, and the optimization will be carried out as further work.

**Table 2.** CPU cycles of enhanced pqsigRM compared with the finalists

| Security | Enhanced pqsigRM | Verification Cycles | | | | |
|---|---|---|---|---|---|---|
| | | Avg | Median | CRYSTALS-DILITHIUM | FALCON | SPHINCS+ |
| 128 | `Enh-pqsigRM-612` | 791,755 | 759,248 | 327,362 | 82,340 | 308,774 |
| 256 | `Enh-pqsigRM-613` | 7,123,917 | 6,951,035 | 871,609 | 168,498 | 696,980 |

# 5 Design Rationale

## 5.1 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 $\lambda$ 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 matrices $\mathbf{H}$, $\mathbf{S}$, and $\mathbf{Q}$. 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 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 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 $\sigma_p^1$ and $\sigma_p^2$. When $p = n/4$, which implies that $\sigma_p^1$ and $\sigma_p^2$ 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 3. 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 386, and 562.

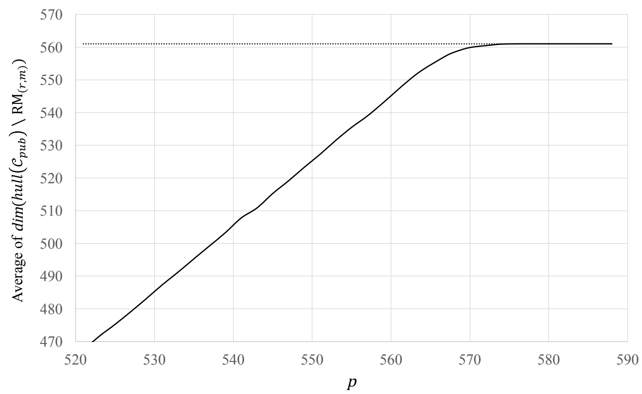


**Fig. 3.** Dimension of $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(6,12)}$ for 128-bit security parameters.

**Table 3.** Average dimension of $hull(\mathcal{C}_{pub})$ and $hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)}$ with $p = n/4$

| $(r,m)$ | (6,12) | (6,13) |
|---|---|---|
| $n$ | 4096 | 8192 |
| $k$ | 2511 | 4097 |
| $dim(hull(\mathcal{C}_{pub}))$ | 1236 | 2974 |
| $dim(hull(\mathcal{C}_{pub}) \setminus \mathrm{RM}_{(r,m)})$ | 386 | 562 |

## 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 [10] can also be applied to RM code-based cryptosystem with insertion. However, we will show below that our proposed algorithm is secure against the above attack methods.

**Security Against Minder-Shokrollahi's Attack** enhanced pqsigRM can be proven to be secure against this attack in the same way with pqsigRM in [7].

**Security Against Chizhov-Borodin's Attack** enhanced pqsigRM can be proven to be secure against this attack in the same way with pqsigRM in [7].

**Security Against Square Code Attack** enhanced pqsigRM can be proven to be secure against this attack in the same way with pqsigRM in [7].

**Other Cryptanalyses of the Signature Scheme Based on Modified RM Code** **1) An attack that finds punctured/inserted elements using hull of public code**

In public key encryption or digital signature algorithm using punctured RM code with random insertion, the punctured/inserted position in the public code and its dual intersection, i.e., hull, is revealed by the fact that they have no support [14, 15]. enhanced pqsigRM can be proven to be secure against this attack in the same way with pqsigRM in [7].

**2) Attack using the probability of 1 for each element of signature**

If the punctured RM code with random insertion is used as a public key, there is a higher probability that the inserted elements of signature to be 1 [13]. In case of puncturing/insertion, the punctured/inserted elements of the error vector are newly computed and replaced in order to generate an error vector with the same syndrome for the shortened/lengthened parity check matrix. Then, it can be used as a signature. In this case, the probability of 1 in the replaced part is relatively high because the other part of the signature is part of the coset leader of the RM code, which is less probably 1. enhanced pqsigRM can be proven to be secure against this attack in the same way with pqsigRM in [7].

**3) Attack using the probability of 1 for each element of near minimum weight codewords**

In public key encryption or digital signature algorithms using punctured RM code with random insertion, the inserted part has a larger average Hamming weight in the near minimum weight codeword of the public code [16]. This attack is also valid because the RM code has greater than or equal to $2^{mr-r(r-1)}$ minimum weight codes [5]. enhanced pqsigRM can be proven to be secure against this attack in the same way with pqsigRM in [7].

### 6.2 Security Analysis

**Decoding One Out of Many**

*Problem 1.* (DOOM)

**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 [17], 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. There are more explanations in [12].

**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. In the enhanced pqsigRM, the syndrome is given as $\mathbf{s} = h(M|i)$, and thus it is also secure against key substitution attacks. There are more explanations in [12].

**EUF-CMA Security**

**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 $\lambda$.*

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 $\mathbf{He}^T = \mathbf{s}_i^T$.

There are more explanations and proof of EUF-CMA security in [12]. Considering these, we obtain the parameters for each security level as in Table 4.

**Table 4.** Parameters for each security level

| $\lambda$ (security) | 128 | 256 |
|---|---|---|
| $(r, m)$ | (6,12) | (6,13) |
| $n$ | 4096 | 8192 |
| $k$ | 2511 | 4097 |
| $w$ | 495 | 1370 |
| $p$ | $\geq$386 | $\geq$562 |
| $k_{rep}$ | 62 | 62 |
| $k_{app}$ | 2 | 2 |

### 6.3 Indistinguishability of Code and Signature in the Proposed Scheme

**Modifications of Public Code** 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 [18], 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 [18] 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 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 $\mathrm{RM}_{(r,r)}$ is replaced with random $(2^r, k_{rep})$ codes** We note that by replacing repeated $\mathrm{RM}_{(r,r)}$ 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 $(\mathrm{RM}_{(r,r)}|\ldots|\mathrm{RM}_{(r,r)})$, 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 $(\mathrm{RM}_{(r,r)}|\ldots|\mathrm{RM}_{(r,r)})$. That is, $(\mathrm{RM}_{(r,r)}|\ldots|\mathrm{RM}_{(r,r)})$

causes the dual code of the partially permuted RM code to have only codewords of even Hamming weight. By replacing the repeated $\mathrm{RM}_{(r,r)}$ 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 odd Hamming weight.

Clearly, the dual code of $\mathrm{RM}_{(r,r)}$ is $\{\mathbf{0}\}$. We replace $\mathrm{RM}_{(r,r)}$ 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 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 Code 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 1.** *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 $\sigma_p^1$ and $\sigma_p^2$.

**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 [19] 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 Summary or 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, enhanced pqsigRM signature scheme has advantages on signature size and verification time. It has the smallest signature size compared with the other digital signature finalist algorithms. Also, it has a very short verification time for 128-bit security. Moreover, the security level is controllable by the parameter setting. 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 $(n-k) \times k$. Besides, it has relatively large number of verification cycles for 256-bit security. For 128 bits of classical security, the signature size of the proposed signature scheme is 512 bytes, which corresponds to 1/4.7 of that of CRYSTALS-DILITHIUM, and the number of median verification cycles is 759,248, which corresponds to the twice of that of CRYSTALS-DILITHIUM.

## 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)

## 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. A. Otmani and H. T. Kalachi, “Square code attack on a modified Sidelnikov cryptosystem,” in *Proc. C2SI*, 2015, pp. 173–183.
11. Knuth, *Seminumerical Algorithms.* The Art of Computer Programming. 2 (3rd ed.). Boston: Addison–Wesley. pp. 145-–146. ISBN 0-201-89684-2. OCLC 38207978.
12. 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.
13. R. Perlner, OFFICIAL COMMENT: pqsigRM, 2018. [Online]. Available: https://gro ups.google.com/a/list.nist.gov/d/msg/pqc-forum/zeHJEzWdv2Y/rABP4ixXAgAJ, [Accessed: 01-Oct-2018]
14. R. Perlner, OFFICIAL COMMENT: pqsigRM, 2018. [Online]. Available: https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/zeHJEzWdv2Y/6eSzqwU4AQAJ, [Accessed: 01-Oct-2018]
15. R. Perlner, OFFICIAL COMMENT: pqsigRM, 2018. [Online]. Available: https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/3l4UmEJFi6k/MeccVRjPAAAJ, [Accessed: 01-Oct-2018]
16. R. Perlner, OFFICIAL COMMENT: pqsigRM, 2018. [Online]. Available: https://groups.google.com/a/list.nist.gov/d/msg/pqc-forum/zeHJEzWdv2Y/6eSzqwU4AQAJ, [Accessed: 01-Oct-2018]
17. N. Sendrier, “Decoding one out of many,” *International Workshop on Post-Quantum Cryptography*, Springer, Berlin, Heidelberg, 2011.
18. T. Debris-Alazard, N. Sendrier, and J.-P. Tillich. “SURF: A new code-based signature scheme,” *arXiv preprint*, arXiv:1706.08065, 2017.
19. 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.

20. 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.
21. 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.
22. 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.