A Modified pqsigRM: RM Code-Based Signature Scheme

Yongwoo Lee, Wijik Lee, Young-Sik Kim, and Jong-Seon No

Abstract

We propose a novel signature scheme based on a modified Reed–Müller (RM) code, which reduces the signing complexity and key size compared to existing code-based signature schemes. This scheme is called as the modified pqsigRM, and corresponds to an improvement of pqsigRM, the proposal submitted to NIST. Courtois, Finiasz, and Sendrier (CFS) proposed a code-based signature scheme using the Goppa codes based on a full domain hash approach. However, owing to the properties of Goppa codes, the CFS signature scheme has drawbacks such as signing complexity and large key size. We overcome these disadvantages of the CFS signature scheme using partially permuted RM code and its decoding, which finds a near codeword for any received vector. Using a partially permuted RM code, the signature scheme resists various known attacks on the RM code-based cryptography. Additionally, we further modify the RM codes by row insertion/deletion of the generator matrix and thereafter resolve the problems reported in the post-quantum cryptography forum by NIST, such as the Hamming weight distribution of the public code.

Index Terms

Code-based cryptography, Courtois, Finiasz, and Sendrier (CFS) signature scheme, digital signature, post-quantum cryptography (PQC), Reed-Müller (RM) code

I. Introduction

RECENTLY, code-based cryptographic algorithms have been extensively studied as a post-quantum cryptography (PQC). McEliece first proved the hardness of the syndrome decoding problem [18], and proposed a cryptosystem based on Goppa codes [21]. In 2001, the Courtois, Finiasz, and Sendrier (CFS) signature scheme was proposed [2]. The CFS signature scheme has certain drawbacks in terms of parameter scaling and a huge signing complexity.

In the CFS signature scheme, an average of $t!$ hashings and decodings are required for signing a message when an (n, k) Goppa code with error correction capability $t = \frac{n-k}{\log n}$ is used. Hence, t should be a relatively small value to reduce the signing time. However, the complexity of a decoding attack is only a polynomial function of the key size with small power, $\approx \text{keysize}^{t/2}$. Thus, the key size should be increased significantly for security. Additionally, with small t, the code rate becomes high and a high-rate Goppa code can be distinguished from a random code [3]. This falsifies the assumption of existential unforgeability under a chosen message attack (EUF-CMA) security proof in [16], based on the indistinguishability of Goppa codes. Although Morozov et al. claim that the strong EUF-CMA security of the CFS signature scheme without the indistinguishability of Goppa codes [17] is proved, their bad parameters remain a drawback of the CFS signature scheme.

In order to solve the problems of the CFS signature scheme, we replace the Goppa code with a modified Reed–Müller (RM) code, where its decoder can decode any received vector into not the closest but somewhat close codeword. In general, an exact error correction is not necessary in code-based signature schemes. It is sufficient to find an error vector with a sufficiently small Hamming weight for a given syndrome and the number of iterations to find a decodable syndrome in the CFS signature scheme can be reduced by such an approach. Signature schemes using such an approach were suggested in [5] and [7]. [5] did not suggest new code or corresponding decoding. Instead, to keep the Hamming weight small enough, this signature scheme uses a sparse coset element added to a small-Hamming-weight codeword and multiplies it by a specific matrix. It is efficient and has a small key size, but an attack algorithm is known [4]. The attack algorithm for the signature scheme in [7] has also been proposed in [6].

Y. Lee and J.-S. No are with the Department of Electrical and Computer Engineering, INMC, Seoul National University, Seoul, 08826, Korea.
W.-J. Lee is with Samsung Electronics, Hwasung, 18448, Korea.
Y.-S. Kim is with the Department of Information and Communication Engineering, Chosun University, Gwangju, 61452, Korea. Y.-S. Kim is the corresponding author. Email: iamyskim@chosun.ac.kr.

There is a well-known decoder for the RM codes, named recursive decoding [14], [8], and this decoder finds a codeword reasonably close to any given received vector. In this respect, the RM code-based signature scheme is a solution for efficient code-based signature scheme. However, the simple replacement of the Goppa code with the RM code results in vulnerability to several known attacks, such as the Minder–Shokrollahi attack [11] and the Chizhov–Borodin attack [10].

Thus, we propose a new code-based signature scheme by using a modified RM code with a partially permuted generator matrix with row insertions and deletions, referred to as a modified pqsigRM. The proposed signature scheme not only overcomes the drawbacks of the CFS signature scheme, but also resists known attacks on RM code-based cryptographic algorithms. By replacing the Goppa code in the CFS signature scheme with a modified RM code, the required key size and signing time are significantly reduced. We also propose a new decoding algorithm for the modified RM code. The decoding method does not guarantee exact error correction but returns a near codeword for any given received vector. This implies that for any given syndrome, we can always find an error vector with a small Hamming weight.

The signature scheme proposed in this work is an improved version that resolves the problems of early versions of pqsigRM [1] submitted to the NIST PQC standardization. In the early versions, column puncturing and insertion were applied to the parity check matrix of the original RM code. Therefore, post-processing was performed on the result of the original RM code decoding, to make it suitable for the secret column puncturing and insertion. Although the column puncturing and insertions are devised to prevent known attacks on RM code-based cryptographic algorithms, a new attack algorithm that finds punctured columns and inserted positions was proposed in the NIST PQC forum. Instead of column insertions and puncturing, we propose an efficient signature scheme using a modified RM code with partial permutation and row insertion/deletion, which finds an error vector with a small Hamming weight for any given syndrome. As a result, we are able to construct a new pqsigRM scheme that is secure both against the attack found during the first round of standardization by NIST, as well as previously known attacks.

For some code-based cryptosystems, side channel attacks using leakage from decoding are known such as in [26]. The algorithm in this work is efficient and constant time algorithm. In addition, since the number of iterations required is significantly reduced, signing process can be implemented at constant time. PQC has a lot of applications such as real-time applications [25]. This algorithm is simple to operate and highly parallelizable, enabling efficient hardware implementation.

The rest of this paper is organized as follows. In Section II, we present concepts of code-based signature scheme and RM codes. The definition of partially permuted RM codes are given in Section III and the proposed signature scheme is presented in Section IV. A method to make the public key indistinguishable from a random matrix is provided by row insertions and deletions of a generator matrix is proposed in Section V. In Section VI, the security of the proposed signature scheme is analyzed, including the EUF-CMA security. The paper is concluded in Section VII.

II. Preliminaries

A. CFS Signature Scheme

The CFS signature scheme is an algorithm created by applying the full domain hash methodology to the Niederreiter cryptosystem. The CFS signature scheme is based on Goppa codes, as in the McEliece cryptosystem, which is given in Algorithm 1.

As described in Algorithm 1, the signing process iterates until a decodable syndrome is found. 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 small to reduce the number of iterations, and high-rate Goppa codes should thus be used. Regarding the key size, the complexity of the decoding attack on the CFS signature scheme is known as small power of the key size, $\approx \text{keysize}^{t/2}$. Hence, the key size should be fairly large to meet a certain security level. In summary, unfortunately, the CFS signature scheme is insecure and inefficient with Goppa codes.

B. Reed–Müller Code and Its Recursive Decoding

RM codes were introduced by Müller and Reed [22], [23]. A decoding algorithm, called recursive decoding, was also proposed in [8]. There are a few definitions for RM codes, but we focus here on recursive definitions of RM codes and recursive decoding using this structure.

The RM code $\text{RM}(r, m)$ is a linear binary $(n = 2^m, k = \sum_{i=0}^{r}\binom{m}{i})$ code, where r and m are integers. $\text{RM}(r, m)$ is defined as $\text{RM}(r, m) = \{(u|u + v)|u \in \text{RM}(r, m-1), v \in \text{RM}(r-1, m-1)\}$, where

Algorithm 1 CFS signature scheme [2]

Key Generation:

$h(\cdot)$ denotes a cryptographic hash function
H is the parity check matrix of an (n, k) Goppa code
The error correction capability t is $\frac{n-k}{\log n}$
S and Q are an $(n-k) \times (n-k)$ scrambler matrix and $n \times n$ permutation matrix, respectively
Public key: $H' = SHQ$ and t
Private key: $H, S,$ and Q

Signing:

M is a message to be signed
$i = 1$
Do
$i \leftarrow i + 1$
Find syndrome $s = h(h(M)|i)$ and compute $s' = S^{-1}s$
Until a decodable syndrome s' is found
Find an error vector satisfying $He'^T = s'$
* Compute $e^T = Q^{-1}e'^T$ and then signature is (M, e, i)

Verification:

Check $\mathrm{wt}(e) \leqslant t$ and $H'e^T = h(h(M)|i)$
If True, then return ACCEPT, else return REJECT

RM$(0, m) := \{(0, 0, \ldots, 0), (1, 1, \ldots, 1)\}$ with code length 2^m and RM$(m, m) := \mathbb{F}_2^{2^m}$. This is a Plotkin's construction and its generator matrix is given as

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

where $G(r, m)$ is the generator matrix of RM(r, m).

The recursive decoding is a soft-decision decoding algorithm that depends on the recursive structure of the RM codes and is described in detail in Algorithm 2, where $y' \cdot y''$ denotes the component-wise multiplication of vectors y' and 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, y'', the second half of the received vector y, is componentwise-multiplied by y', the first half of the received vector. Then, a codeword from RM $(r, m-1)$, i.e., u is removed from y'' and only v and the error vector are left. This is considered as a codeword of RM$(r-1, m-1)$ added to an error vector and reffered as $\hat{v}$. Using $\hat{v}$, we can remove the codeword of RM $(r-1, m-1)$ from the second half of received vector. y' is then added to $y'' \cdot \hat{v}$ and the sum is divided by 2. This is considered as a codeword of RM$(r, m-1)$ added to the error vector and then the decoding is performed. In the recursive decoding, the received vector is further divided into a shorter length, such as $n/4, n/8$, and so on. Finally, we reach RM(m, m) or RM$(0, m)$ and then the division is stopped. Clearly, the minimum distance (MD) decoding of RM(m, m) or RM$(0, m)$ is trivial. The decoding for the entire code is performed by reconstructing these results into the $(u|u+v)$ form.

III. Partially Permuted RM Code and Its Decoding

In this section, we propose a new code, called a partially permuted RM code, and its decoder that decodes any given received vector. Even if an undecodable syndrome is given, the decoding algorithm finds an error vector with a reasonably small Hamming weight.

A. Partial Permutation and Generator Matrix of Partially Permuted RM Code

To construct the generator matrix of the partially permuted RM code, we permute the columns of submatrices of $G(r, m)$. An example of the generator matrix of the code is given in Figure 1. We define partial permutation as a permutation that randomly permutes only some of the columns. Let σ_p^1 and σ_p^2 be two independent partial permutations that randomly permute only p columns out of $n/4$ columns. To generate σ_p^1 and σ_p^2, p elements are

Algorithm 2 Recursive decoding of RM code [8]

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(m, m)
  else
    $(y'|y'') \leftarrow y$
    $y^v = y' \cdot y''$
    $\hat{v} \leftarrow$ RECURSIVEDECODING$(y^v, r-1, m-1)$
    $y^u \leftarrow (y' + y'' \cdot \hat{v})/2$
    $\hat{u} \leftarrow$ RECURSIVEDECODING$(y^u, r, m-1)$
    Output $(\hat{u}|\hat{u} \cdot \hat{v})$
  end if
end function

randomly selected from the index set $\{0, 1, \ldots, n/4 - 1\}$ and the selected elements are randomly permuted, while others are not. Then, σ_p^1 is used to permute the submatrices of $G(r, m)$ corresponding to $G(r, m-2)'s$ in the first row and σ_p^2 is used to permute $G(r-2, m-2)$ in the last row, as in Figure 1. The generator matrix for this partially permuted RM code is shown in Figure 1.

B. Decoding of Partially Permuted RM Code

Complete decoding is applied to the CFS signature scheme [2], where it returns the closest codeword for any given vector. If complete decoding is possible, a coset leader can be found for any syndrome. In other words, for a code with the error correction capability t, complete decoding can succeed even if $t + \delta$ errors exist. In general, it is difficult to design a code with a complete decoder, which is adequate for a code-based signature scheme.

In the CFS signature scheme, complete decoding for the Goppa codes is implemented as follows: the addition of δ errors is equivalent to adding δ randomly chosen columns from the parity check matrix to the syndrome. Because there are $n - k$ linearly independent columns and δ can be a large number, adding δ columns can be any $(n - k)$-tuple binary vector. Hence, adding δ columns to the syndrome is also equivalent to adding a random vector to the syndrome. Clearly, adding a random vector to the syndrome is equivalent to the generation of a new random syndrome, which is implemented as $s = h(h(M)|i)$ by updating a counter i. However, because the Goppa code decoder can still correct errors with Hamming weight less than or equal to t, the probability that the random syndrome is decoded is $\frac{\Sigma_{i=1}^{t}\binom{n}{i}}{n^t} \simeq \frac{1}{t!}$. Therefore, approximately $t!$ syndromes need to be generated and decoded for successful signing.

For the proposed partially permuted RM code, which is simple, scalable, and resistant to structure attacks, there is an algorithm for decoding any received vector. Although this decoding algorithm finds a codeword fairly close to the received vector, it does not always find the nearest codeword. Therefore, the Hamming weight of the decoded error may be large in some cases. Because a valid signature should have a small Hamming weight to prevent forgery, new syndromes are generated until an error vector with a Hamming weight less than or equal to a certain value w is found. Note that the decoding algorithm efficiently finds an error vector with a small Hamming weight for any given syndrome. Hence, it is guaranteed that signature can be successfully generated in dozens to hundreds of iterations, as described in Subsection IV-D. Additionally, because adversaries do not know the decoding algorithm, information set decoding attacks are the best attack algorithms.

The difference between the CFS signature scheme and pqsigRM in decoding is as follows. In the CFS signature scheme, the Hamming weight of the error vector is less than or equal to t and thus the iterations are performed until a decodable syndrome is found. However, in pqsigRM, the signer can decode all syndromes, but repeats until an error with a small Hamming weight w, which may be larger than t, is found to prevent forgery.

When the subcode of the RM code is replaced with its own permutation, the entire code can also be decoded by slightly modified recursive decoding [14]. Moreover, no decoding failure occurs. This is because the recursion eventually reaches RM$(0, m)$ or RM(m, m) and these codes are the MD-decodable codes. We can say that $c = (u|u+v)$ for all $c \in$ RM(r, m), where $u \in$ RM$(r, m-1)$ and $v \in$ RM$(r-1, m-1)$. Recursively, RM$(r, m-1)$ and

$G(r, m-2)^{\sigma_p^1}$	$G(r, m-2)^{\sigma_p^1}$	$G(r, m-2)^{\sigma_p^1}$	$G(r, m-2)^{\sigma_p^1}$
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 generator matrix $G_{\mathcal{M}}$ of RM code $\mathrm{RM}(r, m)$ for the proposed signature scheme.

$\mathrm{RM}(r-1, m-1)$ are also $(u|u+v)$-structured codes, except for $r = 0$ 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 performed, the entire code c can also be decoded [14].

When a code is decodable, its permutation is always decodable when the permutation is known. From this point of view, we define the partially permuted RM code as $\{(u|u+v)|u \in U, v \in V\}$, where $V = \{(u|u+v)|u \in \mathrm{RM}(r, m-2)^{\sigma_p^1}, v \in \mathrm{RM}(r-1, m-2)\}$ and $U = \{(u|u+v)|u \in \mathrm{RM}(r-1, m-2), v \in \mathrm{RM}(r-2, m-2)^{\sigma_p^2}\}$. Here, σ_p^1 and σ_p^2 are partial permutations. The proposed signature scheme uses the partially permuted RM code or modified RM code, as described later, instead of the Goppa code of the CFS signature scheme.

A new decoding algorithm for this code is given in Algorithm 3. The decoding of partially permuted RM codes is possible from the recursive structure of the RM code. Because a permuted code can be decoded using a sequential process of depermutation-decoding-repermutation, every building block of the partially permuted RM code is decodable.

IV. pqsigRM Signature Scheme With Partial Permutation

In this section, we propose a new code-based signature scheme using the partially permuted RM code rather than the Goppa code. The parameter sets for 128, 192, and 256-bit security levels are given and the numerical method to determine the parameters is suggested. In addition, a constant-time signing algorithm is proposed to prevent side channel attacks.

A. Partially Permuted pqsigRM Signature Scheme

1) Key Generation: We randomly generate partial permutations σ_p^1 and σ_p^2. Using σ_p^1 and σ_p^2, a partially permuted generator matrix of $\mathrm{RM}(r, m)$ is generated as in Figure 1 and denoted as $G_{\mathcal{M}}$. Thereafter, the dual matrix of the partially permuted generator matrix $G_{\mathcal{M}}$ becomes the parity check matrix $H_{\mathcal{M}}$. Let S be an $(n-k) \times (n-k)$ random nonsingular matrix and Q be an $n \times n$ random permutation matrix. Then, the public key is $H' = SH_{\mathcal{M}}Q$ and the private keys are $H_{\mathcal{M}}, S, Q, \sigma_p^1$, and σ_p^2.

Using partially permuted RM codes, the proposed signature scheme resists known attacks on RM code-based cryptographic algorithms, such as the Minder–Shokrollahi and Chizhov–Borodin attacks. The subcode to be permuted is designed to keep the dimension of the hull of the code generated by $G_{\mathcal{M}}$ large and it enables the system to resist the known attacks on $(u|u+v)$-structured codes [13].

Algorithm 3 Decoding for partially permuted RM code

function PARTPERMDEC(y, r, m)
  $y \leftarrow y^{\sigma^{-1}}$
  if $r = 0$ **then**
    Perform MD decoding on RM$(0, m)$
  else if $r = m$ **then**
    Perform MD decoding on RM(m, m)
  else
    $(y'|y'') \leftarrow y$
    $y^v = y' \cdot y''$
    $\hat{v} \leftarrow$ PARTPERMDEC$(y^v, r-1, m-1)$
    $y^u \leftarrow (y' + y'' \cdot \hat{v})/2$
    $\hat{u} \leftarrow$ PARTPERMDEC$(y^u, r, m-1)$
    $y \leftarrow (\hat{u}|\hat{u} \cdot \hat{v})$
  end if
  Output y^σ
end function
* σ is σ_p^1 or σ_p^2 for permuted block and identity, otherwise.

TABLE I: Parameters for each security level

	security	(r, m)	(n, k)	w	p
pqsigRM-5-11	128	(5, 11)	(2048, 1024)	306	129
pqsigRM-6-12	192	(6, 12)	(4096, 2510)	467	385
pqsigRM-6-13	256	(6, 13)	(8192, 4096)	1400	561

2) Signing: To sign a given message M, randomly choose i from $\{0,1\}^{n-k}$. A binary vector $s = h(h(M|H')|i)$ is calculated, where $h : \{0,1\}^* \rightarrow \{0,1\}^{n-k}$ is a cryptographic hash function. Our goal is to find the error vector e satisfying $H'e^T = SH_{\mathcal{M}}Qe^T = s$. Let us take $s' = S^{-1}s$. Performing the decoding as in Algorithm 3, we find an error vector e' satisfying $H_{\mathcal{M}}e'^T = s'$. If $\mathrm{wt}(e') \leqslant w$, we compute $e^T = Q^{-1}e'^T$, and the signature is then given as (M, e, i). Otherwise, the decoding process is repeated by choosing another i. It is worth noting here that unlike the CFS signature scheme, updating i is not due to decoding failure, but to find a syndrome that can be decoded into an error vector with a Hamming weight less than or equal to w.

3) Verification: If $\mathrm{wt}(e) \leqslant w$ and $H'e^T = h(h(M|H')|i)$, we return ACCEPT and REJECT, otherwise.

The key generation, signing, and verification processes are described in Algorithm 4.

B. Parameter Sets

The parameters of the proposed signature scheme consist of $r, m, w,$ and p. It should be noted that $n, k,$ and d_{min} are derived directly from r and m as $2^m, \sum_{i=0}^{r}\binom{m}{i}$, and $2^{(m-r)}$, respectively. As given in Table I, the parameter sets of the partially permuted pqsigRM signature schemes are proposed for 128, 192, and 256-bit security levels. These are referred to as pqsigRM-5-11, pqsigRM-6-12, and pqsigRM-6-13, respectively, and their generator matrices are given in Figure 2. It is noted that the larger the weight parameter w, the faster the signing time, but the smaller w, the higher the security level. Here, w is derived to satisfy the desired security level based on the attack algorithm by Stern [12], which is described in Section VI-A. It is also noted that the smaller the value of p, the shorter the signing time, but the lower the randomness. If p is large, the signing time increases. We can see that some characteristics of the codes are retained by lowering p to a certain extent, which will be described in Section IV-C.

C. Statistical Analysis for Determining Parameter p

The hull of a code is defined as the intersection of the code and its dual code. The public code of the proposed signature scheme refers to the code that its parity check matrix is the public key of the proposed signature scheme.

$G(5,9)^{\sigma_p^1}$	$G(5,9)^{\sigma_p^1}$	$G(5,9)^{\sigma_p^1}$	$G(5,9)^{\sigma_p^1}$
0	$G(4,9)$	0	$G(4,9)$
0	0	$G(4,9)$	$G(4,9)$
0	0	0	$G(3,9)^{\sigma_p^2}$

(a) Partially permuted generator matrix $G_{\mathcal{M}}$ for 128-bit security level.

$G(6,10)^{\sigma_p^1}$	$G(6,10)^{\sigma_p^1}$	$G(6,10)^{\sigma_p^1}$	$G(6,10)^{\sigma_p^1}$
0	$G(5,10)$	0	$G(5,10)$
0	0	$G(5,10)$	$G(5,10)$
0	0	0	$G(4,10)^{\sigma_p^2}$

(b) Partially permuted generator matrix for 192-bit security level.

$G(6,11)^{\sigma_p^1}$	$G(6,11)^{\sigma_p^1}$	$G(6,11)^{\sigma_p^1}$	$G(6,11)^{\sigma_p^1}$
0	$G(5,11)$	0	$G(5,11)$
0	0	$G(5,11)$	$G(5,11)$
0	0	0	$G(4,11)^{\sigma_p^2}$

(c) Partially permuted generator matrix for 256-bit security level.

Fig. 2: Generator matrices of partially permuted RM codes.

Algorithm 4 Partially permuted pqsigRM signature scheme

Preprocessing:

For a given partially permuted (n, k) RM code and the security level, derive w for successful signing as in Table I.

Key Generation:

Generate random partial permutations σ_p^1 and σ_p^2
Using σ_p^1 and σ_p^2, generate a partially permuted generator matrix $G_{\mathcal{M}}$
Generate $H_{\mathcal{M}}$ from $G_{\mathcal{M}}$
Generate S and Q
Compute $H' = SH_{\mathcal{M}}Q$
Private key: $H_{\mathcal{M}}, S, Q, \sigma_p^1$, and σ_p^2
Public key: H', w

Signing:

M is a message to be signed
Do
$i \xleftarrow{R} \{0,1\}^{n-k}$
Find syndrome $s = h(h(M|H')|i)$ and compute $s' = S^{-1}s$
Perform decoding and find an error vector e' satisfying $H_{\mathcal{M}}e'^T = s'$
Until e' is found such that $\mathrm{wt}(e') \leqslant w$
* Compute $e^T = Q^{-1}e'^T$ and then signature is (M, e, i)

Verification:

Check $\mathrm{wt}(e) \leqslant w$ and $H'e^T = h(h(M|H')|i)$
If True, then return ACCEPT, else return REJECT

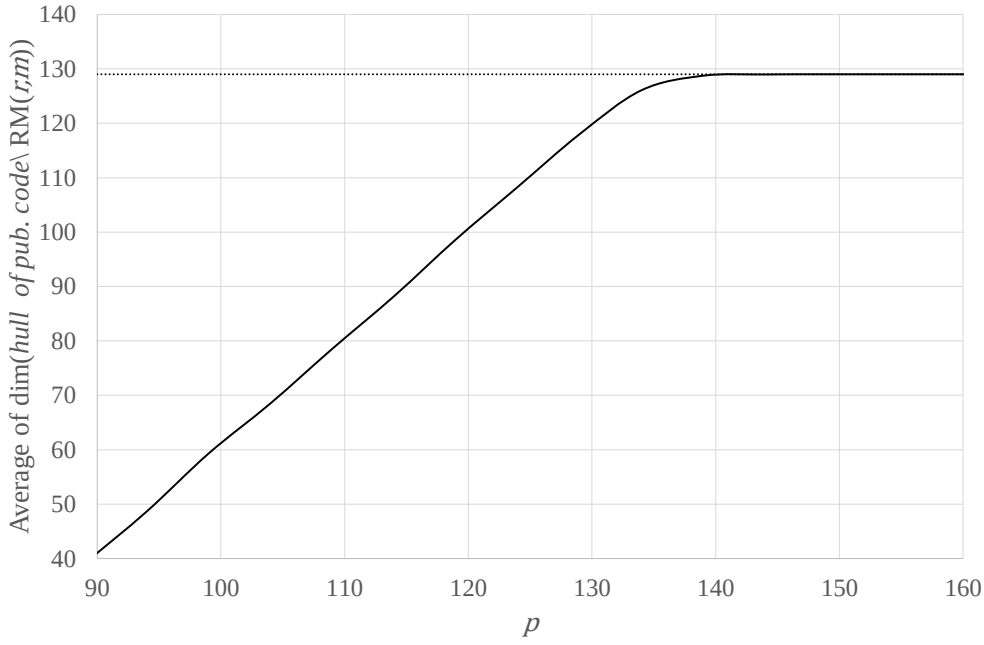
TABLE II: Dimension of *hull of partially permuted pqsigRM's public code*$\setminus$RM(r, m) with $p = n/4$

	(n, k)	dim(hull)	dim(*hull*$\setminus$*RM*)
pqsigRM-5-11	(2048, 1024)	766	129
pqsigRM-6-12	(4096, 2510)	1236	385
pqsigRM-6-13	(8192, 4096)	2974	561

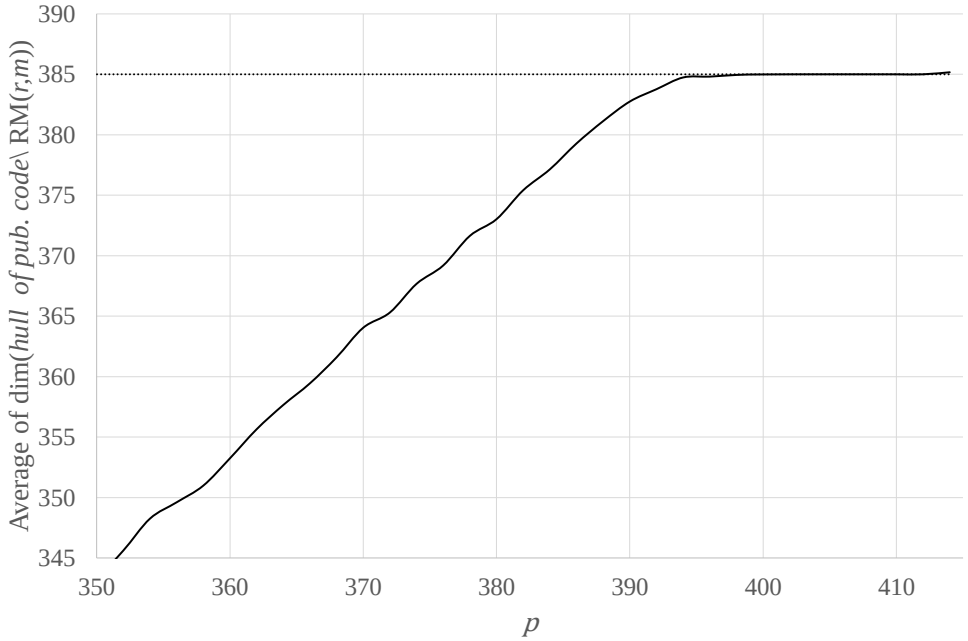
For the partially permuted RM code, its hull is overlapped with (not a subset of) the original RM code. If the hull is a subset of the original RM code and its dimension is large, the minimum-Hamming-weight codeword of the original RM code may be included in the hull. Then, the Minder–Shokrollahi attack might be applied using such minimum-Hamming-weight codewords. Therefore, to prevent the Minder–Shokrollahi and Chizhov–Borodin attacks, the hull of the public code should not be a subset of the original RM code and *hull of public code*$\setminus$RM(r, m) *permuted by* Q must occupy a large portion of the hull, where $\setminus$ denotes the relative complement.

Because the permutation Q is not important to the analysis for determining the parameter p, we ignore Q in this subsection, and permutation implies the partial permutations σ_p^1 and σ_p^2. When p is $n/4$, which means that σ_p^1 and σ_p^2 are full permutations, the dimension of the hull and the dimension of *hull of public code*$\setminus$RM(r, m) are given in Table II. The value of the dimensions in Table II might be slightly changed according to the permutation.

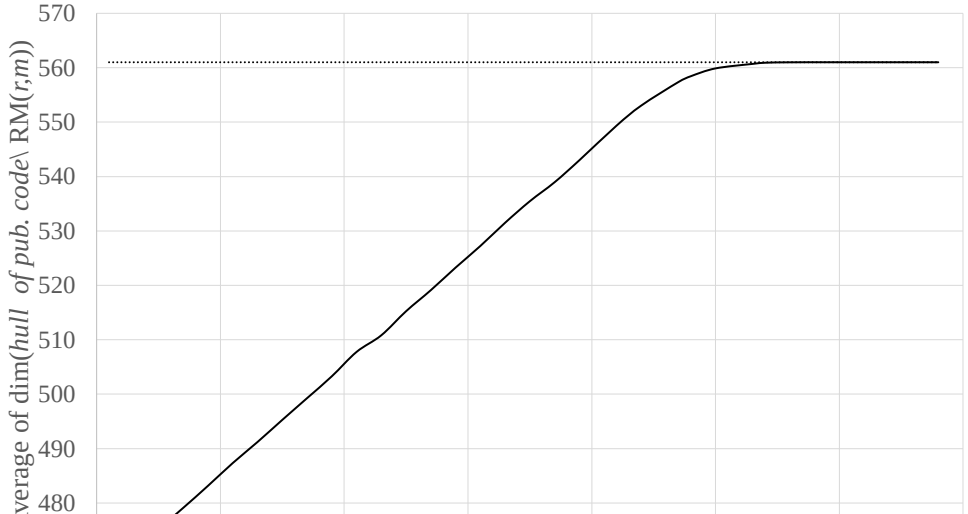
If p is small, the Hamming weight of the errors decreases. Hence, the signing time can be reduced by using partial permutation with p rather than full permutation. We want to find a smaller value for p maintaining the dimension of *hull of public code*$\setminus$RM(r, m) as large as with full permutation. We can see that the average of the dimension of *hull of public code*$\setminus$RM(r, m) tends to increase as p increases and is saturated when p is above a certain value in Figure 3. More specifically, the dimension of *hull of public code*$\setminus$RM(r, m) is saturated when p is the approximately the average dimension of *hull of public code*$\setminus$RM(r, m) for full permutation. Hence, for pqsigRM-5-11, pqsigRM-6-12, and pqsigRM-6-13, we determine p as 129, 385, and 561, respectively, as in Table I.



(a) Dimension of *hull of public code*$\setminus$RM$(5, 11)$ for pqsigRM-5-11.



(b) Dimension of *hull of public code*$\setminus$RM$(6, 12)$ for pqsigRM-6-12.



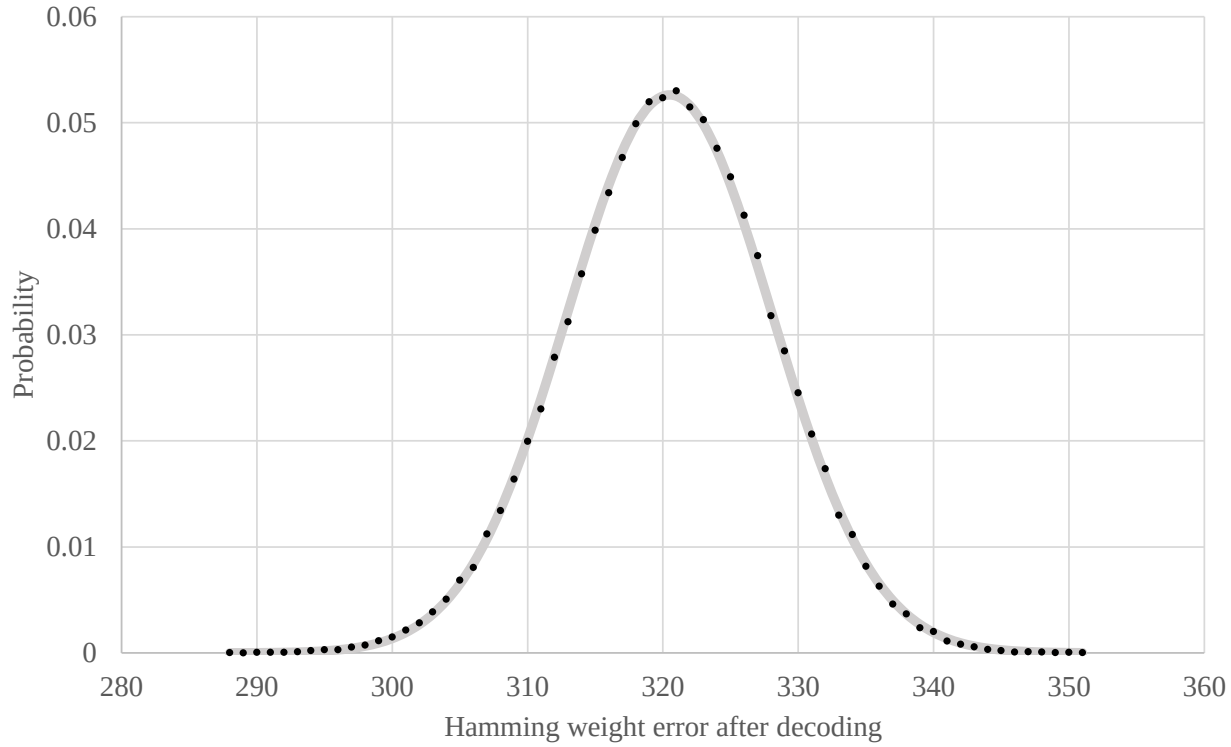


Fig. 4: Distribution of Hamming weights of decoded results among 10^5 trials for pqsigRM-5-11. The gray line shows a normal distribution with the same mean and variance.

D. Number of Iterations for the Signature and Constant-Time Algorithm

Let N be the number of iterations for decoding that guarantees successful signing. In the proposed signature scheme, signatures can be generated within only a few iterations. However, in this case, the time taken for signature is not a constant time. To prevent side channel attack due to the non-constant signing time, the number of iterations for decoding is fixed to N, even when the successful signature is obtained before N iterations. In this section, we find the number of iterations that guarantees successful signing. Because the recursive decoding algorithm is a constant-time algorithm, we can design a constant-time signing algorithm for the proposed signature scheme with a fixed number of iterations.

The signing is successful if the Hamming weight of the error vector from decoding the partially permuted RM code is less than or equal to w for the hashed message with a random number i, $s = h(h(M|H')|i)$. Letting X_j be the Hamming weight of the error vector for the jth counter, the probability of successful signing is given by

$$\Pr\left\{\min_{j\leqslant N}(X_j) \leqslant w\right\} = 1 - (\Pr\{X_i > w\})^N, \tag{1}$$

where each X_j is assumed to be independent and identically distributed. The probability $\Pr\{X_i > w\}$ can be numerically obtained by the distribution of Hamming weights of the decoded results. For 128-bit security, the distribution of Hamming weights of the decoded errors is numerically obtained as in Figure 4. Also, the probability of finding a valid signature in a single trial and the value of N that guarantees successful signing can be numerically found as listed in Table III.

Let us compare the minimum number of iterations that guarantees successful signing in the proposed signature scheme and the CFS signature scheme. Let us consider the security of the CFS signature scheme to be approximately given by $2^m t$, and using the fact that $t = \frac{n-k}{\log n}$, we can derive some parameter sets for 128-bit security as in Table IV. The first option in the table is to make the key size similar to pqsigRM. In this case, the number of iterations is 10^9, which is impossible to implement. However, $t = 9$ is suggested in [2], which is the second option in the table.

TABLE III: Parameter N that guarantees successful signing

	w	success prob. of single trial	target failure probability	N
pqsigRM-5-11	306	0.02788	$\leqslant 2^{-128}$	3138
pqsigRM-6-12	467	0.13833	$\leqslant 2^{-192}$	894
pqsigRM-6-13	1400	0.99997	$\leqslant 2^{-256}$	18

TABLE IV: Minimum number of iterations that guarantees successful signing in CFS signature scheme with 128-bit security level

	(n, k)	key size	t	N
CFS	$(2^{11}, 2^{11} - 121)$	512 KB	11	$\geqslant 10^9$
	$(2^{14}, 2^{14} - 126)$	32 MB	9	$\geqslant 10^7$
	$(2^{25}, 2^{25} - 125)$	128 TB	5	$\geqslant 10^4$
pqsigRM	$(2^{11}, 2^{10})$	256 KB	307	3138

There are still many more iterations that are required to ensure success and the key size is hundreds of bytes. The last option is to match the number of iterations with pqsigRM. Here, we can see that the key size is infeasible. With the CFS signature scheme, it is very difficult to build a signature scheme that guarantees the successful signing in a constant time, but it can be implemented in a reasonable time with pqsigRM. As shown in Table III, successful signing is even guaranteed within a few dozen times in certain parameter sets.

V. Indistinguishable Variants of pqsigRM

The proof of the EUF-CMA security of the partially permuted pqsigRM requires the indistinguishability of the public code of pqsigRM from a random code. Hence, we will discuss how to design an indistinguishable public code of pqsigRM in this section.

Any $(u|u+v)$-structured code used in code-based cryptosystems should have a high-dimensional hull for security, and the public code of pqsigRM should be designed to have a $(u|u+v)$ structure for efficient decoding without failure. The partially permuted RM code has a $(u|u+v)$ structure, which enables recursive decoding. As shown in [13], for any $(u|u+v)$-structured code, i.e., $\{(u|u+v)|$ all $u \in U,$ all $v \in V\}$, when $U^{\perp} \cap V = \{0\}$, the hull of the public code is highly probable to have a $(u|u)$ structure, where $\perp$ denotes the dual code. This $(u|u)$ structure reveals information of the permutation Q. To avoid such a disadvantage, we should maintain the high dimension of $U^{\perp} \cap V$, which means that the dimension of the hull of the public code is high. Hence, we design the public code of pqsigRM as indistinguishable from a random code of which the hull dimension is the same as that of the public code, rather than any random linear code.

The *Goppa Code Distinguishing Problem* is a key underlying problem for the security of the McEliece cryptosystem. In general, the indistinguishability is assumed to be hold, but it has not been proved. Instead, the Hamming weight distribution of the Goppa code is considered as evidence for the indistinguishability [19]. It is certain that the Hamming weight distribution of the Goppa code is a binomial distribution, which is the Hamming weight distribution of a random code.

To design the public code of pqsigRM having a binomial Hamming weight distribution, we perform three modifications on its generator matrix as follows:

i) A random row with odd Hamming weight is appended to the generator matrix.
ii) A random codeword chosen from its dual code is added as a row of the generator matrix.
iii) One row of the generator matrix other than the above two added rows is removed to replace the subcode.

These modifications are assumed to make the public code indistinguishable from a random code whose hull dimension is the same as that of the public code.

A. Appending a Row With Odd Hamming Weight

The Hamming weight distribution of the public code of pqsigRM should be a binomial distribution, which is the Hamming weight distribution of a random code. However, the partially permuted RM code has only codewords with even Hamming weight. This is because the Hamming weights of codewords of RM (r, m) are even numbers,

except for $r = m$, and the codewords of the permuted RM codes are the same. From a numerical analysis, we can see that the Hamming weight distribution of the public code of pqsigRM is a binomial distribution of even values.

By appending a random row with odd Hamming weight to the generator matrix, we make the Hamming weight distribution of the public code a binomial distribution. This requires a slight modification of the signing process. By adding a random row with odd Hamming weight to the generator matrix, the code dimension k is increased by one and the decoding should be modified. Let c_{odd} be the appended random row with odd Hamming weight and C be the public code before adding c_{odd}. By appending c_{odd}, the code becomes $C + \{0, c_{odd}\}$, i.e., $C \sqcup (C + c_{odd})$. The decoding is performed for both the received vector y and $y + c_{odd}$. That is, the decoding algorithm performs PARTPERMDEC in Algorithm 3 twice and returns the error with a smaller Hamming weight.

B. Adding a Random Codeword of Dual Code

The Hamming weights of codewords in the hull of the partially permuted RM code are only multiples of 4. However, codewords in the hull of a random code have even Hamming weight, not only multiples of 4. Similar to Subsection V-A, a random codeword is appended to the hull. By appending a random codeword to the hull, we force codewords of the hull of the public code to have even Hamming weights, not only multiples of 4.

Appending a codeword to the hull is more complicated than appending a codeword to the code. The following procedure explains how to append a random codeword to the hull. Let C_{hull} be the hull of a code C. Then, we define C' and C'' by $C = C_{hull} + C'$ and $C^{\perp} = C_{hull} + C''$, where C_{hull}, C', and C'' are linearly independent subsets. We can then generate a code with a hull dimension of $dim(C_{hull}) + 1$ by the following procedure:

i) Find a codeword $c_{dual} \in C''$ such that $c_{dual} \cdot c_{dual} = 0$. It is easy to find such codeword, because a codeword with even Hamming weight satisfies $c_{dual} \cdot c_{dual} = 0$.
ii) Let $C_{inc} = C + \{c_{dual}\} = (C_{hull} + \{c_{dual}\}) + C'$.
iii) Because $c_{dual} \cdot (C_{hull} + \{c_{dual}\}) = \{0\}$ and $c_{dual} \cdot C' = \{0\}$, $c_{dual} \in C_{inc}^{\perp}$, where for a vector x and a set of vectors A, $x \cdot A$ is the set of inner products of x and elements of A.
iv) We can see that $C_{inc} \cap C_{inc}^{\perp} = (C_{hull} + \{c_{dual}\})$. Hence, C_{inc} is a code that has a hull dimension of $dim(C_{hull}) + 1$.

Using the above method, we can generate codewords in the hull of pqsigRM with even Hamming weight, not only multiples of 4. A numerical analysis shows that the Hamming weight distribution of codewords in the hull follows a binomial distribution of even values. If the Hamming weights of the codewords of the hull are only multiples of 4, another c_{dual} is selected and the process is repeated.

C. Reducing Dimension of Public Code by One

The codewords of the dual code of the partially permuted RM code have only an even Hamming weight. This is due to a subcode of the partially permuted RM code, and we solve this problem by replacing this subcode with another code whose MD decoder exists. The public code includes $(\mathrm{RM}(r,r)|\ldots|\mathrm{RM}(r,r))$, whose dual code has only codewords of even Hamming weight, as given in theorem below. It is trivial that the dual code of the public code is a subset of the dual code of $(\mathrm{RM}(r,r)|\ldots|\mathrm{RM}(r,r))$. This means that the repeated part $(\mathrm{RM}(r,r)|\ldots|\mathrm{RM}(r,r))$ causes the dual code to have only codewords of even Hamming weight.

Theorem 1. *Let C be a code whose dual code has only codewords of even Hamming weight. Then, the dual code of a concatenated code of two codes, $\{(c|c)|c, c \in C\}$, has only codewords of even Hamming weight.*

Proof. Let $h \in (C|C)^{\perp}$, where C is an (n,k) code and $C|C$ is a concatenated code given as $\{(c|c)|c \in C\}$. We define vectors of length n, h_1, and h_2 by $h = (h_1|h_2)$. It is obvious that if $h_1 \in C^{\perp}$, then $h_2 \in C^{\perp}$. If $h_1 \notin C^{\perp}$, we have $h_1 \cdot c + h_2 \cdot c = 0$, i.e., $h_1 \cdot c = h_2 \cdot c$. This implies that $h_1 = h_2$. Hence, $\mathrm{wt}(h)$ is even. □

By replacing the repeated $\mathrm{RM}(r,r)$ with another code whose dual code has codewords of odd Hamming weight, we can force the dual of the public code to have odd-Hamming-weight codewords.

By definition, the dual code of $\mathrm{RM}(r,r)$ is $\{0\}$. We replace $\mathrm{RM}(r,r)$ with a (n',k') code, which can be decoded by MD decoding. A simple example is given, where $k' = n' - 1$. The MD decoding of this code can be performed by the following process:

i) Let $\{0, h = (h_1, \ldots, h'_n)\}$ be the dual of an $(n', n'-1)$ code.
ii) For a received vector y, select an index such that $h_i = 1$ and then the nearest codeword is $y - (0, \ldots, h_i, \ldots, 0)$ when the syndrome of y is not zero.

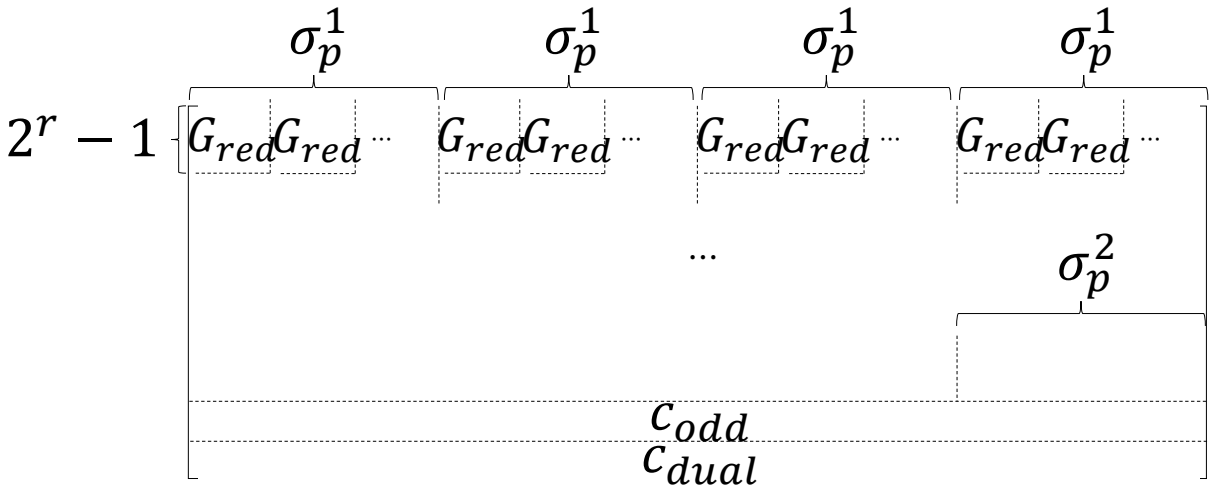


Fig. 5: Generator matrix of the modified RM code.

iii) y is a codeword, otherwise.

We can generate an $(n', n'-1)$ random code for our purpose by generating a random vector h with odd Hamming weight. Assume that $h = (1, h_2, \ldots, h'_n)$, which corresponds to the parity check matrix. Then, its generator matrix is given as

$$G_{red} = \left[(h_2, \ldots, h'_n)^T \quad | \quad I_{n'-1}\right].$$

The generator matrix is easy to find even if $h_1 = 0$.

The RM code with all the modifications in Subsections V-A, V-B, and V-C, together with partial permutation is called the modified RM code. The generator matrix of the modified RM code is described in Figure 5.

D. New Decoding Algorithm

To apply the above-mentioned modification to the public code, the processes of the proposed signature scheme, pqsigRM, should be changed accordingly to adapt to the modified decoding algorithm. Let H_{mod} be the parity check matrix of the modified RM code. Then, the public key H' is given by $H' = SH_{mod}Q$ and its decoding algorithm is given in Algorithm 5.

VI. Security Analysis of Modified pqsigRM

In this section, the security of the modified pqsigRM is analyzed. The information set decoding, which is the most general and well-known attack on the code-based cryptosystem is first considered. Thereafter, we show that the modified pqsigRM is resistant to known attacks on cryptosystems based on the RM code, owing to the partial permutation and modifications. Finally, it is shown that the modified pqsigRM is EUF-CMA secure.

A. Analysis of Information Set Decoding Attack

Information set decoding is a brute-force attack method that finds the error vector e such that $He^T = s$ and $\mathrm{wt}(e) \leqslant w$, where Stern improved the attack complexity [12]. Although more efficient information set decoding methods than Stern's algorithm have been proposed, Stern's algorithm is the most universal and simple to use. Thus, the security of the modified pqsigRM against information set decoding is analyzed based on Stern's algorithm.

The complexity of Stern's algorithm is given as

$$C_{Stern}(g, l) := Kl\binom{k/2}{g/2}$$

and the success probability when there is one and only one such error vector is

$$Pr_{Stern}(g, l) := \frac{\binom{k/2}{g/2}^2 \binom{n-k-l}{w-g}}{\binom{n}{w}},$$

Algorithm 5 Decoding for modified RM code

function DECODE(y)
  $ErrSet \leftarrow \{(y+c) - \text{MODDEC}(y+c, r, m)$
    $|c \in \{0, r_{odd}, r_{dual}, r_{odd}+r_{dual}\}\}$
  Output $\text{argmin}_{e \in ErrSet}\, \text{wt}(e)$
end function

function MODDEC(y, r, m)
  $y \leftarrow y^{\sigma^{-1}}$
  if $r = 0$ **then**
    Perform MD decoding on RM$(0, m)$
  else if $r = m$ **then**
    Perform MD decoding on RM(m, m)
     or using G_{red}
  else
    $(y'|y'') \leftarrow y$
    $y^v = y' \cdot y''$
    $\hat{v} \leftarrow \text{MODDEC}(y^v, r-1, m-1)$
    $y^u \leftarrow (y' + y'' \cdot \hat{v})/2$
    $\hat{u} \leftarrow \text{MODDEC}(y^u, r, m-1)$
    $y \leftarrow (\hat{u}|\hat{u} \cdot \hat{v})$
  end if
  Output y^σ
end function
*σ is σ_p^1 or σ_p^2 for permuted block and identity, otherwise.
*For the replaced subcode as in Figure 5, MD decoding is performed using G_{red}.

TABLE V: Work factor for the proposed parameters

	$(n, k, d_{\min}, w)$	optimal g	WF
pqsigRM-5-11	(2048, 1024, 32, 306)	26	2^{128}
pqsigRM-6-12	(4096, 2510, 64, 467)	64	2^{192}
pqsigRM-6-13	(8192, 4096, 128, 1400)	136	2^{280}

where $l = \log\binom{k/2}{g/2}$, g is an integer that can be adjusted to minimize the complexity, and the hidden parameter K is considered as $\frac{\log n}{2}$ for the actual computation. However, in the modified pqsigRM, there are many n-tuple error vectors with Hamming weight less than or equal to w for each syndrome. The number of n-tuple error vectors with Hamming weight less than or equal to w for a given syndrome is approximately given by $\frac{\sum_{i=0}^{w}\binom{n}{i}}{2^{n-k}} \simeq \frac{\binom{n}{w}}{2^{n-k}}$. Hence, the success probability of Stern's algorithm is approximately given as

$$Pr_{Stern}(g,l)\frac{\binom{n}{w}}{2^{n-k}} = \frac{\binom{k/2}{g/2}^2\binom{n-k-l}{w-g}}{2^{n-k}}.$$

Dividing complexity of Stern's algorithm WF by the probability above, the computational complexity for Stern's algorithm is given by

$$WF = Kl\frac{2^{n-k}}{\binom{k/2}{g/2}\binom{n-k-l}{w-g}}.$$

The value of WF for each parameter set of the modified pqsigRM is given in Table V.

B. Analysis of Known Attacks

The Minder–Shokrollahi [11] and Chizhov–Borodin [10] attacks are well-known attacks for RM code-based cryptosystems, which decompose the public key $H' = SHQ$ into the private keys S, H, and Q. In addition, the

square code attack [9] can also be applied to RM code-based cryptosystems with random column insertion. As the modified pqsigRM does not rely on random column insertion or code puncturing, we do not need to consider attacks that target punctured or inserted RM codes, such as the square code attack. We show that our modified pqsigRM algorithm is secure against the Minder–Shokrollahi and Chizhov–Borodin attacks.

1) Minder–Shokrollahi Attack: One of the major objects of the attack on the McEliece cryptosystem is to find the permutation matrix Q. Let $\mathrm{RM}(r,m)^Q$ be the partially permuted code of $\mathrm{RM}(r,m)$ for some unknown permutation Q. In the Minder–Shokrollahi attack, the attack procedure to find a permutation Q consists of three steps as follows:

i) Find codewords in $\mathrm{RM}(r,m)^Q$ that belong to $\mathrm{RM}(r-1,m)^Q$. It is required to find enough such codewords to build a basis of $\mathrm{RM}(r-1,m)^Q$.
ii) Iterate the previous step until $\mathrm{RM}(1,m)^Q$ is obtained.
iii) Find a permutation η such that $\mathrm{RM}(1,m)^{Q\cdot\eta} = \mathrm{RM}(1,m)$. This implies $\eta = Q^{-1}$. Then, we have $\mathrm{RM}(r,m)^{Q\cdot\eta} = \mathrm{RM}(r,m)$.

It is clear that the first step is crucial for the success of this attack. Let $x \in C$ be a minimum-Hamming-weight codeword.

The minimum-Hamming-weight codewords are used in the first step of the Minder–Shokrollahi attack. The algorithm to find $\mathrm{RM}(r-1,m)^Q$ from $\mathrm{RM}(r,m)^Q$ is based on the following proposition.

Proposition 2 ([11]). *Let $f \in RM(r,m)$ be a minimum-Hamming-weight codeword. Then, there exist $f_1, f_2, \ldots, f_r \in RM(1,m)$ such that*

$$f = f_1 \cdot f_2 \cdots f_r,$$

where the f_i are the minimum-Hamming-weight codewords of $RM(1,m)$ in function form.

However, the properties of the minimum-Hamming-weight codewords are different because of the partially permuted submatrix of the generator matrix. Therefore, Proposition 2 is not true for the modified RM code with partially permuted generator matrix.

Because the minimum Hamming weight of the original RM code is 2^{m-r}, one might consider attempting the same attack by finding the codeword with Hamming weight 2^{m-r} of the public code. In fact, among the minimum-Hamming-weight codewords of the public code, there are codewords that are not affected by partial permutation. For example, as we can see in the generator matrix shown in Subsection II-B, the codewords with two repetitions of minimum-Hamming-weight codewords of $\mathrm{RM}(r,m-1)$ are not only codewords of the original RM code, but also of the public code, and are not affected by the partial permutation. However, the number of such codewords is only approximately $1/2^r$ of all the minimum-Hamming-weight codewords of the original RM code. Furthermore, because there are many more codewords than the minimum-Hamming-weight codewords and half of the elements of each codeword are modified, the Hamming weight distribution of codewords has been randomly changed.Therefore, it is difficult to find codewords with Hamming weight 2^{m-r} satisfying Proposition 2. Thus, enough independent codewords to generate $\mathrm{RM}(r-1,m)^Q$ cannot be found, owing to the partial permutation of the RM code.

In conclusion, the modified pqsigRM is resistant to the Minder–Shokrollahi attack, because it cannot perform the process of finding $\mathrm{RM}(r-1,m)$ in $\mathrm{RM}(r,m)$.

2) Chizhov–Borodin Attack: From an RM code $\mathrm{RM}(r,m)$, $\mathrm{RM}(2r,m)$ can be constructed with low polynomial-time complexity. Similarly, $\mathrm{RM}(kr,m)$ can easily be constructed. Moreover, $\mathrm{RM}(m-r-1,m)$, a dual code of $\mathrm{RM}(r,m)$, can also be constructed in low polynomial-time complexity. Thus, $\mathrm{RM}(kr+l(m-1),m)$ can be obtained and finally, we have $\mathrm{RM}(\gcd(r,m-1),m)$. If $\gcd(r,m-1)=1$, then $\mathrm{RM}(1,m)$ is directly found. Otherwise, $\mathrm{RM}(r-1,m)$ can be obtained by the Minder–Shokrollahi attack. By iterating this procedure until we have $\gcd(r-k,m-1)=1$, $\mathrm{RM}(1,m)$ can be found. It is then straightforward to find the permutation η, that is, Q^{-1} [10].

However, the dual of the public code of the modified pqsigRM is not an RM code. Moreover, the algorithm to generate $\mathrm{RM}(r_1+r_2,m)$ using $\mathrm{RM}(r_1,m)$ and $\mathrm{RM}(r_2,m)$ is not applicable to the public code, because it is not an RM code. Therefore, the Chizhov–Borodin attack is not applicable to the modified pqsigRM.

C. Security Against Key Substitution Attack

An attack that finds a valid key different from the correct key that satisfies the verification for a message signature pair is called a key substitution attack. If the adversary knows the private key and the public key corresponding to the message and signature pair, it is called a weak-key substitution attack, and if he knows only the public key, it is called a strong-key substitution attack. Both polynomial-time weak- and strong-key substitution attacks on the CFS

signature scheme were proposed in [20]. A modification that resists the attack was also proposed in [20]. In this modification, the syndrome s is generated by hashing the message, counter, and public key, rather than only hashing the message and counter. It is shown that this modified CFS signature scheme is secure against key substitution attacks [17]. In the modified pqsigRM, the syndrome is given as $s = h(h(M|H')|i)$, and therefore it is also secure against key substitution attacks.

D. Security Against Existential Forgeries

In this subsection, we prove the EUF-CMA security of the modified pqsigRM by applying Dallot's proof for the EUF-CMA security of the CFS signature scheme [16]. Because the used codes are different, the EUF-CMA security of the modified pqsigRM can be proved differently from that of the CFS signature scheme. However, because the proofs are almost the same, the parts overlapping with the existing one are omitted.

EUF-CMA is a usual attack model against signature schemes. An EUF-CMA is viewed as a game played between the adversary and challenger. The public key PK, hash oracle $\mathcal{H}$, and signing oracle Σ are given to a $(\tau, q_{\mathcal{H}}, q_{\Sigma})$-adversary $\mathcal{A}$, where $\mathcal{A}$ can query $q_{\mathcal{H}}$ hash values and q_{Σ} signatures for inputs of his own choice. Within a maximum τ computation time, $\mathcal{A}$ tries to find a valid message signature pair (m^*, σ^*). $\mathcal{A}$ wins the game if $\text{Verifying}(m^*, \sigma^*, PK) = 1$ and σ^* has not been provided by Σ, and the challenger wins the game, otherwise. A signature scheme is $(\epsilon, \tau, q_{\mathcal{H}}, q_{\Sigma})$-EUF-CMA secure if for any $(\tau, q_{\mathcal{H}}, q_{\Sigma})$-adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the game is less than or equal to ϵ.

In [16], finding an existential forgery of the CFS signature scheme under the CMA has been reduced to the *Goppa Code Distinguishing Problem* and *Goppa Parameterized Bounded Decoding Problem*, which are defined as:

Definition 3. (Goppa Code Distinguishing Problem [16])
Input: *An* $(n-k) \times n$ *parity check matrix* H'
Output: *A bit* $b \in \{0, 1\}$ *where* $b = 1$ *if* $H' = SHQ$*, where* H *is the parity check matrix of a Goppa code,* S *is an* $(n-k) \times (n-k)$ *permutation matrix, and* Q *is an* $n \times n$ *permutation matrix, and* $b = 0$*, otherwise.*

Definition 4. (Goppa Parameterized Bounded Decoding Problem [16])
Input: *An* $(n-k) \times n$ *parity check matrix* H' *and a syndrome* $s \in \mathbb{F}_2^{n-k}$
Output: *A word* $e \in \mathbb{F}_2^n$ *such that* $\text{wt}(e) \leqslant \frac{n-k}{\log n}$ *and* $H'e^T = s$.

However, as an efficient algorithm that distinguishes high-rate Goppa codes from random codes has been proposed, it is shown that the *Goppa Code Distinguishing Problem* is not a hard problem and thus the proof of Dallot is disabled.

A new proof of strong EUF-CMA of the CFS signature scheme that does not rely on the *Goppa Code Distinguishing Problem* is proposed [17]. This proof reduces the strong EUF-CMA security to *Goppa Code Decoding Problem*, defined as:

Definition 5. (Goppa Code Decoding Problem [17])
Input: *An* $(n-k) \times n$ *parity check matrix* $H' = SHQ$ *and a syndrome* $s \in \mathbb{F}_2^{n-k}$*, where* H *is an* $(n-k) \times n$ *parity check matrix of a binary Goppa code,* S *is a nonsingular* $(n-k) \times (n-k)$ *matrix, and* Q *is an* $n \times n$ *permutation matrix*
Output: *A word* $e \in \mathbb{F}_2^n$ *such that* $\text{wt}(e) \leqslant \frac{n-k}{\log n}$ *and* $H'e^T = s$.

However, for any code, including a Goppa code, the decoding problem is more difficult than the distinguishing problem, as given by the following theorem.

Theorem 6. *The Goppa Code Distinguishing Problem is reducible to the Goppa Code Decoding Problem.*

Proof. Assume that $\mathcal{A}_{dec}$ is an adversary that efficiently solves the Goppa Code Decoding Problem. $\mathcal{A}_{dec}$ has inputs H' and s and returns e satisfying $\text{wt}(e) \leqslant \frac{n-k}{\log n}$ and $H'e^T = s$. Now, we can make $\mathcal{A}_{dist}$ an adversary solving *Goppa Code Distinguishing Problem* using $\mathcal{A}_{dec}$. $\mathcal{A}_{dist}$ gives H' and s to $\mathcal{A}_{dec}$ as inputs. Subsequently, if $\text{wt}(\mathcal{A}_{dec}(H', s)) \leqslant t$ and $H'e^T = s$, $\mathcal{A}_{dist}$ returns 1. Else if $\text{wt}(\mathcal{A}_{dec}(H', s)) > t$ or $\mathcal{A}_{dec}(H, s) = \perp$ or $H'e^T \neq s$, it returns 0. □

Although it has been proved that the *Goppa Code Distinguishing Problem* is not a hard problem for specific parameters proposed by CFS, i.e., a high-rate Goppa code, the proof in [16] is tighter than it is in [17]. Therefore, we prove the EUF-CMA security of the modified pqsigRM by modifying the proof in [16].

Now, we reduce the EUF-CMA security of the modified pqsigRM to the *Modified RM Code Distinguishing Problem* and the *Bounded Decoding With Hull of Large Dimension*, where two definitions are given as:

Definition 7. (Modified RM Code Distinguishing Problem)
Input: *An* $(n-k)\times n$ *parity check matrix* H'
Output: *A bit* $b\in\{0,1\}$ *where* $b=1$ *if* $H'=SH_{mod}Q$*, where* H_{mod} *is the parity check matrix of a modified RM code,* S *is an* $(n-k)\times(n-k)$ *nonsingular matrix, and* Q *is an* $n\times n$ *permutation matrix, and* $b=0$*, otherwise.*

Definition 8. (Bounded Decoding With Hull of Large Dimension)
Input: *An* $(n-k)\times n$ *parity check matrix* H' *and a syndrome* $s\in\mathbb{F}_2^{n-k}$*, where the dimension of the hull of the code corresponding to* H' *is large (approximately* $\geqslant\frac{n}{4}$*)*
Output: *A word* $e\in\mathbb{F}_2^n$ *such that* $\mathrm{wt}(e)\leqslant\frac{n-k}{2}$ *and* $He^T=s$*.*

It is difficult to prove the hardness of the *Modified RM Code Distinguishing Problem*. In fact, because it is extremely difficult to prove the hardness of distinguishing a code from a random code, several cryptosystems are designed assuming that their public codes are difficult to distinguish from random codes. Based on the Hamming weight distribution of codewords, as described in Section V, we claim that the *Modified RM Code Distinguishing Problem* is a hard problem. For successful decoding for any received vector, a $(u|u+v)$-structured code should be utilized; to resist the attack on the $(u|u+v)$-structured code in [13], we design the code with a high hull dimension. Generally, the dimension of the hull of a random code is very low, but it is shown that there are numerous codes with a certain dimension of hull in [24]. Therefore, only with the dimension of the hull, it is hard to know whether the code is random one with high dimensional hull or the public code of the modified pqsigRM.

The *Goppa Parameterized Bounded Decoding Problem* belongs to the general decoding problem. The general decoding problem is proved to be NP-hard for $t\leqslant\frac{n-k}{2}$ in the binary case [18]. The problem in Definition 8 is the addition of the hull dimension condition to the general decoding problem. In [18], the general decoding problem is reduced to *Three-Dimensional Matching*, which is a well-known NP-hard problem. However, dimension of the hull is not considered in the proof in [18] and it is difficult to determine how the dimension of the hull affects the problem. We thus assume it to be a hard problem. Furthermore, a definition of (τ,ϵ)-hard is given as follows.

Definition 9. ((τ,ϵ)-hard)
A problem is said to be (τ,ϵ)*-hard if for any solver running in time at most* τ*, its success probability is less than or equal to* ϵ*.*

Now, we can modify the theorem in [16] and prove the EUF-CMA security of the modified pqsigRM as follows.

Theorem 10. *The modified pqsigRM is* $(\epsilon,\tau,q_{\mathcal{H}},q_\Sigma)$*-EUF-CMA secure in the random oracle model under the assumption that the Modified RM Code Distinguishing Problem and Bounded Decoding With Hull of Large Dimension are* $(\tau_{dist},\epsilon_{dist})$*- and* $(\tau_{decode},\epsilon_{decode})$*-hard, respectively. Here,* ϵ *and* τ *are given as*

$$\begin{aligned}\epsilon =&(q_{\mathcal{H}}+q_\Sigma+1)\epsilon_{decode}+\epsilon_{dist}\\&+2-(1-\frac{1}{2^{n-k}})^{(q_{\mathcal{H}}+q_\Sigma+1)}-(1-\frac{q_\Sigma}{2^{n-k}})^{q_{\mathcal{H}}}\end{aligned}$$

and

$$\tau\geqslant\tau_{decode}-(q_{\mathcal{H}}+q_\Sigma+1)\cdot T_s(n,k),$$

where $T_s(n,k)$ *is the syndrome computation time for an* (n,k) *modified RM code.*

Proof. The proof is similar to that in [16] and is omitted. □

For quantum EUF-CMA security, the security reduction in [15] can be adapted with the same assumptions as above.

VII. Conclusion

We introduced a new signature scheme based on modified RM codes with partial permutation that improves the CFS signature scheme. For any given syndrome, an error vector with small Hamming weight can be found from the structure of the partially permuted RM code. The proposed signature scheme resists known attacks against cryptosystems based on the original RM codes. The partially permuted RM code improves the signature success

condition in the CFS signature scheme and can greatly shorten the signing time. In addition, errors larger than the error correction capability can be detected through decoding, and thus the key size can be reduced.

We further modified the RM code with row insertion/deletion, which shows the indistinguishability from random codes with the same hull dimension while maintaining the decoding of the partially permuted RM code. Based on this indistinguishability assumption and the hardness of decoding a random code whose hull dimension is large, we proved the EUF-CMA security of the proposed signature scheme. The challenge of proving that these two assumptions are true will be addressed in the future.

REFERENCES

[1] W. Lee, Y.-S. Kim, Y. Lee, and J.-S. No, "Post quantum signature scheme based on modified Reed-Muller code pqsigRM," *First round submission to the NIST post quantum cryptography call,* November 2017.

[2] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *Asiacrypt,* Gold Coast, Australia, 2001, pp. 157-174.

[3] 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.

[4] A. Phesso and J.-P. Tillich, "An efficient attack on a code-based signature scheme," in *PQCrypto,* Fukoka, Japan, 2016, pp. 86-103.

[5] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani, "Using LDGM Codes and sparse syndromes to achieve digital signatures," in *PQCrypto,* Limoges, France, 2013, vol. 7932, pp. 1-15.

[6] D. Moody and R. Perlner, "Vulnerabilities of "McEliece in the World of Escher", in *PQCrypto,* Fukuoka, Japan, 2016, pp. 104-117.

[7] D. Gligoroski, S. Samardjiska, H. Jacobsen, and S. Bezzateev, "McEliece in the world of Escher," *IACR Cryptology ePrint Archive,* Report2014/360, 2014, http://eprint.iacr.org/.

[8] 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.

[9] A. Otmani and H. T. Kalachi, "Square code attack on a modified Sidelnikov cryptosystem," in *Proc. C2SI*, 2015, pp. 173-183.

[10] 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).

[11] L. Minder and A. Shokrollahi, "Cryptanalysis of the Sidelnikov cryptosystem," in *Eurocrypt* 2007, LNCS, vol. 4515, 2007, pp. 347-360.

[12] J. Stern, "A method for finding codewords of small weight," *Coding Theory and Applications*, vol. 388, pp. 106-133, 1989.

[13] T. Debris-Alazard, N. Sendrier, and J.-P. Tillich, "The problem with the SURF scheme," *arXiv preprint arXiv:1706.08065,* 2017.

[14] F. Hemmati, "Closest coset decoding of $u|u+v|$ codes," *IEEE J. Sel. Areas Commun.*, vol. 7, pp. 982-988, Aug. 1989.

[15] A. Chailloux and T. Debris-Alazard, "A tight security reduction in the quantum random oracle model for code-based signature schemes," *arXiv preprint arXiv:1709.06870,* 2017.

[16] L. Dallot, "Towards a concrete security proof of Courtois, Finiasz, and Sendrier signature scheme," in *Proc. WEWoRC*, vol. 4945, 2007, pp. 65-77.

[17] 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.

[18] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384-386, May 1978.

[19] M. Finiasz, "Words of minimal weight and weight distribution of binary Goppa codes," in *IEEE International Symposium on Information Theory,* Yokohama, Japan, 2003, p. 70.

[20] 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.

[21] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report,* vol. 4244, pp. 114-116, 1978.

[22] 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.

[23] 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.

[24] N. Sendrier, "On the dimension of the hull," *SIAM Journal of Discr. Math.,* vol. 10, no. 2, pp. 282-293, May 1997.

[25] A. A. Yavuz, A. Mudgerikar, A. Singla, I. Papapanagiotou, and E. Bertino, "Real-time digital signatures for time-critical networks," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 11, pp. 2627-2639, Nov 2017.

[26] C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt, "Horizontal and vertical side channel analysis of a McEliece cryptosystem," IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1093-1105, June 2016.