Post quantum signature scheme based on modified Reed-Muller code
pqsigRM

Principal submitter This submission is from the following team.

- Wijik Lee, Seoul National University
- Young-Sik Kim, Chosun University
- Yong-Woo Lee, Seoul National University
- Jong-Seon No, Seoul National University

E-mail address: jsno@snu.ac.kr
Telephone: +82-10-5241-3450 Postal address: Department of Electrical and Computer Engineering, Seoul National University, #011, 1, Gwanak-ro, Gwanak-gu, Seoul, 08826, Rep. of KOREA

Auxiliary submitters: There are no auxiliary submitters. The principal submitter is the team listed above.

Inventors/developers: The inventors/developers of this submission are the same as the principal submitter. Relevant prior work is credited below where appropriate.

Owner: Same as submitter.

Signature: See also printed version of "Statement by Each Submitter".

- Wijik Lee ____________________________
- Young-Sik Kim ____________________________
- Yong-Woo Lee ____________________________
- Jong-Seon No ____________________________

Contents

1 Introduction

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

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). The error correctability 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, 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 submission, we replace the Goppa code with the RM code in the CFS signature scheme. RM code can use complete decoding by utilizing a 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. Further, the proposed signature scheme can compromise the signing time and security level by adjusting the allowable maximum Hamming weight of error vectors, called the error weight parameter $w = t + \delta$.

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 attack [7] and Chizhov–Borodin attack [8]. 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. These attacks can be similarly applied to the RM code-based signature scheme. It is shown herein that punctured RM codes with random insertion can be secure against these attacks, and an optimal puncturing scheme to prevent Minder–Shokrollahi and Chizhov–Bordin attacks is proposed [9]. In addition, it is also shown that the punctured RM code with random insertion is secure from a square code attack [10], which can distinguish randomly inserted columns from the modified generator matrix. It is also proved that the proposed modified RM code-based signature scheme is EUF-CMA secure because the parity check matrix of the modified RM code in the proposed signature scheme is not distinguishable from a random matrix.

2 General algorithm specification (part of 2.B.1)

2.1 Parameter space

This pqsigRM submission provides a code-based signature scheme. Each operation has four parameters: (r, m) are positive integers of RM code parameters, p is a positive integer of puncturing parameter of the RM code, and w is the error weight parameter.

2.2 Private key and public key

1-1) Puncturing with random insertion of generator matrix: Let G and H be a $k \times n$ generator matrix and an $(n-k) \times n$ parity check matrix of the RM code, RM(r, m), respectively. First, convert the generator matrix G to a systematic form. Generally, the generator matrix of RM code cannot be converted to systematic form directly. It can be transformed to the row reduced echelon form (RREF). After RREF process, the generator matrix G is tranformed to G_{RREF}, where $G_{\text{RREF}} = EG$ with successive elementary row operation E. Then we modify the G_{RREF} to the systematic form G_{sys} by multiplying permutation matrix Q_1, where $G_{\text{sys}} = G_{\text{RREF}}Q_1 = [I|P]$. Also, for a parity check matrix H, the following equation holds, that is, $H_{\text{sys}} = [P^T|I]$. Let L_D be a set of indices of puncturing positions in the parity part P of the systematic form of the generator matrix, which was described for the nonsystematic RM code in Algorithm 1. The procedure for puncturing generator matrix and determining the set L_D is described in Algorithm 1, and two important notations for Algorithm 1 are defined as follows.

Definition 1. *The support of a codeword $c \in RM(r, m)$ is defined as the set of indices i such that $c_i \neq 0$, $0 \leq i \leq n-1$, denoted as supp(c).*

Definition 2. *Let c be a codeword of C and L be an index set. Then,* $\text{proj}_L(c)$ *is a sub-codeword composed of the components with indices in L from c. In addition, for a linear code C, we define* $\text{proj}_L(C) = \{\text{proj}_L(c) | c \in C\}$.

Because the puncturing procedure of the generator matrix of RM code is given in Algorithm 1, the parity check matrix H corresponding to G can be modified, similarly.

We set L_D as the last p columns of P in G_{sys}, which are punctured, that is, $L_D = \{n-k-p+1, n-k-p+2, \cdots, n-k\}$. The L_D satisfies the conditions in Algorithm 1. Then, the modification of the parity check matrix is described in Algorithm 2.

Then, the modified parity check matrix H_m can be described as in Fig. 1, where R is a $p \times (n-p)$ binary random sub-matrix with row vectors $r_i = (r_{ij})$, $n-k-p+1 \leq i \leq n-k, 1 \leq j \leq n-p$, and P' is the last p-column deleted version of P.

The deleted and inserted rows are not necessarily the same number as well as the same position but here, we assume that they are the same.

Algorithm 1 [9] Puncturing procedure of generator matrix

```
Input: $k \times n$ generator matrix $G$ of RM code
Output: index set $L_D$
1. Randomly pick a minimum Hamming weight codeword $x$ from $C$.
2. Randomly pick a minimum weight codeword $y$ from $\mathrm{proj}_{\mathrm{supp}(x)}(C)$.
3. Choose $p$ such that $\mathrm{wt}(y) \le p$.
4. Randomly choose the set $L_D$ of indices such that $\mathrm{supp}(y) \subseteq L_D$ and $|L_D| = p$.
```

Algorithm 2 Modification of parity check matrix of punctured RM code in systematic form

```
Input: $k \times n$ generator matrix $G = [I|P]$ of the systematic form of RM code.
Output: modified parity check matrix $H_m$.
1. Let $L_D = \{n-k-p+1, n-k-p+2, \cdots, n-k\}$ be an row index set in the systematic
   form of parity check matrix $H_{\mathrm{sys}} = [P^T|I]$ using $p$ in Algorithm 1.
2. Replace the last $p$ rows of the parity check matrix $H_{\mathrm{sys}}$ by the binary random vectors
   $r_i$, $n-k-p+1 \le i \le n-k$ denoted as $H_m$, where
```

$$r_{ij} = \begin{cases} 1, & \text{for } \; j = i+k \\ 0, & \text{for } \; n-p+1 \le j \le n, j \ne i+k \\ \text{random binary bits, otherwise.} \end{cases}$$

1-2) Generation of S, Q, and H_m: Let S be an $(n-k) \times (n-k)$ scrambling matrix and Q be an $n \times n$ permutation matrix. The public key is generated by calculating $H' = SH_mQ$, and the private keys are S, H_m, Q, R, and Q_1. The binary matrix R is row-wisely generated by a random number generator based on AES-256 (shortly, RNG-AES-256) and Q is generated by Knuth shuffling algorithm [11]. In order to reduce key size, the public key H' is RREF. The set of column indices of leading coefficient and parity part of matrix are stored.

The proposed pqsigRM uses private key v instead of Q_1, where v is the set of indices with size $n-k$ that stores the indices of information columns of H in numerical order. It is equivalent to know Q_1 and v because knowing the indices of information (pivot) columns, we can easily transform H to H_{sys}.

2.3 Signatures

For a message M, choose random integer i generated by RNG-AES-256. Using hash function h, define the syndrome as $s = h(h(M)|i)$, which is the same as that of the CFS signature scheme, where | denotes vertical concatenation. Here the hash function

$$H_m = \begin{array}{c|c} & \begin{array}{ccc} k & n-k-p & p \end{array} \\ \hline \begin{array}{c} n-k-p \\ p \end{array} & \left[\begin{array}{cc:c} P'^T & I_{n-k-p} & 0 \\ \hdashline \multicolumn{2}{c:}{R} & I_p \end{array}\right] \end{array}$$

Figure 1: Modified parity check matrix of the proposed signature scheme.

$$\begin{array}{c|c} & \begin{array}{ccc} k & n-k-p & p \end{array} \\ \hline \begin{array}{c} n-k-p \\ p \end{array} & \left[\begin{array}{cc:c} P'^T & I_{n-k-p} & 0 \\ \hdashline \multicolumn{2}{c:}{R} & I_p \end{array}\right] \end{array} \left[\begin{array}{c} e'_1 \\ \vdots \\ e'_{n-p} \\ \hdashline r_1 \\ \vdots \\ r_p \end{array}\right] = \left[\begin{array}{c} s'_1 \\ \vdots \\ s'_{n-k-p} \\ \hdashline s'_{n-k-p+1} \\ \vdots \\ s'_{n-k} \end{array}\right]$$

Figure 2: Signing process of the proposed signature scheme.

h is SHA-512, where the length of a hash output is 512 bits. Since the length of syndrome s should be longer than 512, we concatenate the output of hashes as follow: $s = h(h(M)|i)|h(h(h(M)|i))|\cdots|h^{(l)}(h(M)|i)$, where $h^{(l)}(\cdot)$ denotes applying hash function l times.

2-1) Find the closest coset: Find the error vector e such that $SHQe^T = s$. Let $e'^T = Qe^T$ and $s' = S^{-1}s$. Set $e'_{v[i]} = s'_i$ for $1 \leq i \leq n-k$, where $v[i]$ denotes the ith index in v. Then, $He'^T = s'$ holds. Perfom complete decoding to find new error vector e'.

2-2) Find the punctured part of the error vector: Because the parity check matrix H is random row-deleted and inserted as H_m, we have to replace the last p elements of e' by $e'_p = [r_1, r_2, \cdots, r_p]$, denoted as $e' = [e'_{n-p}|e'_p]$, such that $H_m e'^T = s'$. Let $s' = [s'^T_{n-k-p}|s'^T_p]^T$, where s'_{n-k-p} and s'_p denote the first $n-k-p$ and last p elements of s', respectively. Then,

$H_m e'^T = s'$ can be rewritten as

$$\begin{bmatrix} [P'^T|I_{n-k-p}]e'^T_{n-p} \\ Re'^T_{n-p} + e'^T_p \end{bmatrix} = \begin{bmatrix} s'_{n-k-p} \\ s'_p \end{bmatrix}.$$

Thus, we have

$$e'^T_p = s'_p + Re'^T_{n-p}$$

and thus $e' = [e'_{n-p}|s'^T_p + e'_{n-p}R^T]$.

If the Hamming weight of e' is larger than the error weight parameter w, then we randomly choose another i and apply the signing process again, where w is larger than the error correctability t. The expected number of iterations is given as $N = 10000$, which will be discussed in the next subsection.

If $\mathrm{wt}(e') \leq w$, compute $e^T = Q^{-1}e'^T$, and the signature σ is then given as $\sigma = (M, e, i)$.

2.4 Verification

Check $\mathrm{wt}(e) \leq w$ and $H'e^T = h(h(M)|i)|h(h(h(M)|i))|\cdots|h^{(l)}(h(M)|i)$. If TRUE, then return ACCEPT; else, return REJECT.

The key generation, signing, verification processes are simplified as Algorithm 3.

3 List of parameter sets (part of 2.B.1)

3.1 Parameter set pqsigRM-4-12

Uses RM code RM(4,12) with $w = 1285, p = 20$.

3.2 Parameter set pqsigRM-6-12

Uses RM code RM(6,12) with $w = 480, p = 20$.

3.3 Parameter set pqsigRM-6-13

Uses RM code RM(6,13) with $w = 1367, p = 30$.

4 Design rationale (part of 2.B.1)

The proposed pqsigRM uses the parameters N and w, where N is the expected number of iterations and w is the error weight parameter. The signing time and security levels are

```
**Algorithm 3** The proposed signature scheme
Preprocessing:
    For a given modified $(n,k)$ RM code and the security level larger than 128 bits, derive
    $(N,w)$ for successful signing as in Table 1.
Key Generation:
    Generate random matrices $S$, $Q$, and $R$.
    Get index set $v$ from $G$.
    Generate $H_m$ as in Algorithm 2.
    Compute $H' = SH_mQ$.
    Private key: $S, Q, R$, and $v$.
    Public key: $H'$, $w$
Signing:
    Do
        Choose random number $i_r$
        Find syndrome $s = h(h(M)|i_r)|h(h(h(M)|i_r))|\cdots|h^{(l)}(h(M)|i_r)$ and compute $s' = S^{-1}s$.
        Set $e'_{v[i]} = s'_i$ for $1 \leq i \leq n-k$. Then, $He'^T = s'$ holds.
        Perform complete decoding and find new $e'$ such that $He'^T = s'$.
        Find $e_p'^T = s'_p + Re_{n-p}^T$ and thus $e' = [e'_{n-p}|e'_p]$.
    Until $\mathrm{wt}(e') \leq w$
    * Compute $e^T = Q^{-1}e'^T$ and thus signature is $\sigma = (M, e, i_r)$.
Verification:
    Check $\mathrm{wt}(e) \leq w$ and $H'e^T = h(h(M)|i_r)|h(h(h(M)|i_r))|\cdots|h^{(l)}(h(M)|i_r)$.
    If True, then return ACCEPT, else return REJECT.
```

Table 1: The probability of successful signing for parameters N and w in RM$(5, 10)$.

$w \setminus N$	10,000	20,000	40,000
90	0.01	0.02	0.04
93	0.1	0.18	0.33
96	0.41	0.83	0.97
97	0.83	0.97	1
98	0.97	1	1
99	1	1	1

Table 2: An error weight parameter w for 10,000 random syndromes.

	$(n, k, d_{\min})$	Error weight parameter w
RM(4, 12)	(4096, 794, 256)	1285
RM(6, 12)	(4096, 2510, 64)	480
RM(6, 13)	(8192, 4096, 128)	1463

determined by w and N. In [12], the method to derive the value of w and N is proposed.

To determine what is the appropriate value of w in the proposed signature scheme, we perform simulations for random syndromes. For N random syndromes s, we find the minimum Hamming weight of error vector e satisfying $H'e^T = s$ by carrying out complete decoding. The required number N of counters i, the corresponding error weight parameter w, and probability of successful signing in the signing stage are listed in Table 1.

Assume that N is the expected number of signing trials for the successful signing in the signing stage. The signing is successful if the complete decoded error weight is less than or equal to w for the hashed message with counter i, $h(h(M)|i)$. Let X_i be the Hamming weight of error vector by the complete decoding for counter i. Then the probability of successful signing is given as

$$\begin{aligned} \text{prob}\left\{\min_{i\leq N}(X_i) \leq w\right\} &= 1 - \text{prob}\left\{X_1 > w, X_2 > w, \cdots, X_N > w\right\} \\ &= 1 - \left(\text{prob}\{X_1 > w\}\right)^N \end{aligned} \tag{1}$$

where each X_i is assumed to be i.i.d. The probability prob$\{X_1 > w\}$ can be numerically obtained by the distribution of Hamming weights of coset leaders in the complete decoding. Thus, N and w can be selected for successful signing for the given RM code using (1). For RM$(5, 10)$, the distribution of Hamming weights of coset leaders is numerically obtained from Fig. 3, where the minimum Hamming weight of error vectors among 10^7 random syndromes is 87. In Table 1, the probability of successful signing for RM(5, 10) is listed for parameters N and w using (1).

We repeat the same procedure on RM(4, 12), RM(6, 12), and RM(6, 13). Table 2 presents

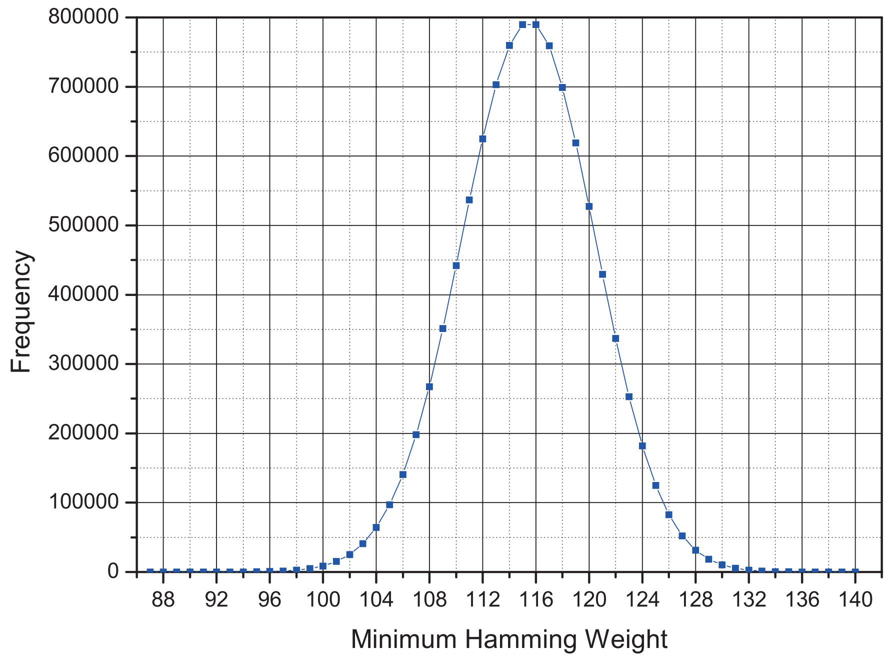

Figure 3: Distribution of Hamming weights of coset leaders among 10^7 in RM$(5, 10)$.

the error weight parameter w for successful signing in $N = 10,000$ iterations. If we simulate for more syndromes, smaller weight of e can be obtained at the cost of longer signing time. That is, if we choose the smaller error weight parameter w and the larger N in the proposed signature scheme, the level of security becomes higher, but the time for successful signing is increased. In Table 2, the error weight parameter w that can be successfully signed with $N = 10,000$ trials and its signing time in a straightforward implementation on —Intel(R) Xeon(R) CPU E5-2698 v4 2.20GHz for each RM code are described.

5 Detailed performance analysis (2.B.2)

5.1 Description of platform

The following measurements were collected using a desk-top computer with CPU is an Intel —Intel(R) Xeon(R) CPU E5-2698 v4 2.20GHz— running at 2.2 GHz. Turbo Boost is disabled. This machine has 128GB of RAM and runs Ubuntu 16.04 LTS. Benchmarks has ran on one core of the CPU. The source code can be compiled by `gcc *.c -lssl -lcrypto -lm`. in directory `/pqsigRM-4-12`, `/pqsigRM-6-12`, `/pqsigRM-6-13`. Since the signing algorithm is a probabilistic algorithm, number of iteration at signing varies. Therefore, we fixed the number of iteration as the mean value of 30 experiments with different messages and keys. number of iterations as the average value of 30. For the detailed descriptions about the success probability of the signing, see 2.B.1.

Table 3: CPU cycles of pqsigRM with —Intel(R) Xeon(R) CPU E5-2698 v4 2.20GHz—

	security	key generation	signing	verification
`/pqsigRM-4-12`	128	9641836	15194705	81178
`/pqsigRM-6-12`	196	1983428	77735436	116906
`/pqsigRM-6-13`	256	22668519	1557210	540378

Table 4: Public key and secret key size of pqsigRM

	public key	secret key
`/pqsigRM-4-12`	336804	1382118
`/pqsigRM-6-12`	501176	334006
`/pqsigRM-6-13`	2144166	2105344

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

5.2 Time

The following measurements are CPU cycler for run `/pqsigRM-4-12`, `/pqsigRM-6-12`, `/pqsigRM-6-13` at —Intel(R) Xeon(R) CPU E5-2698 v4 2.20GHz—. The measurements are given in Table 3

5.3 Space

Sizes are straightforwardly calculated from parameters (and confirmed in various experiments). Specifically, keys are 2^{18} bytes for `pqsigRM-4-12`, 2^{19} bytes for `pqsigRM-6-12`, 2^{22} bytes for `pqsigRM-6-13`, if the transmitted messages are short enough. Signatures are $260, 516$, and 1028 bytes longer. The key sizes are given in Table 4

5.4 How parameters affect performance

The performance can be easily controlled by parameters N and w. You can control the singing time and the difficulty of the attack by changing w, the minimum weight of the error. Decreasing w, the probability that the desired error is obtained is lowered. Therefore, N should increase inversely with w. This makes signing time longer. However, if w increases, the attack probability decreases. The current system uses a fixed value, but by changing w, we can create a digital signature system with variable security. Thus, w, the secrecy, and N, signing time are trade-offs.

5.5 Optimizations

For further optimizations, an efficient implementation of binary matrix operations are required.

6 Expected strength (2.B.4) in general

6.1 Security definitions

The proposed pqsigRM is designed for EUF-CMA security.

6.2 Rationale

See Chapter 8 for an analysis of semantic and known attacks.

7 Expected strength (2.B.4) for each parameter set

7.1 Parameter set pqsigRM-4-12

Category 1, security level satisfies 2^{128}.

7.2 Parameter set pqsigRM-6-12

Category 3, security level satisfies 2^{192}.

7.3 Parameter set pqsigRM-6-13

Category 5, security level satisfies 2^{256}.

Table 5: The security of the proposed signature scheme for $N = 10,000$.

	$(n, k, d_{\min}, w)$	$\frac{\sum_{i=0}^{w} \binom{n-k}{i}}{2^{n-k}} \times C$
RM(4, 12)	(4096, 794, 256, 1285)	$\leq 2^{-128}$
RM(6, 12)	(4096, 2510, 64, 480)	$\leq 2^{-192}$
RM(6, 13)	(8192, 4096, 128, 1367)	$\leq 2^{-256}$

8 Analysis of known attacks (2.B.5)

8.1 Forgery attack

The attacker tries to forge the signature with public key H' and hashed message $h(h(M)|i)$. Assume that the public key H' is systematic, where $H' = [H_0|I]$, I is an $(n-k) \times (n-k)$ identity matrix, and H_0 is an $(n-k) \times k$ matrix. Then, attacker computes z satisfying the following equation

$$H'z^T = [H_0|I][z_1|z_2]^T = s = h(h(M)|i) \tag{2}$$

where z_1 and z_2 are vectors with size k and $n-k$, respectively. The attacker can let z_1 be an all-zero vector and $z_2 = s$. If the Hamming weight of z_2 is less than or equal to w, then the forgery is successful. The probability of successful forgery is given as

$$\frac{\sum_{i=0}^{w} \binom{n-k}{i}}{2^{n-k}}. \tag{3}$$

Table 5 presents the security for each RM code and error weight parameters w of error vectors given in Table 1.

8.2 Information set decoding attack

Information set decoding is brute force method that find the error vector e such that $He^T = s$ and $\mathrm{wt}(e) \leq w$. The algorithm for information set decoding is given in Algorithm 4. Stern [6] optimized the complexity of the information set decoding. The complexity is given as follows.

$$WF = \binom{k/2}{p}\binom{n-k-l}{w-2p} \Big/ \binom{n}{w}$$

The value of WF for each RM code is given in Table 6.

8.3 RM code structure attack

Minder-Shokrollahi attack [7] and Chizhov-Borodin attack [8] are well-known attack for RM code-based cryptosystem which decompose the public key $H' = SHQ$ into the private keys

Algorithm 4 Information set decoding attack [5]

```
• Input: a $k \times n$ matrix $H$, integer $w$
• Output: a non-zero codeword of Hamming weight $\leq w$
    – Pick $n \times n$ permutation matrix $P$
    – Compute $H' = UHP = (I|R)$
    – Compute all the sum of $p$ rows or less of $H'$, if one of those sums has weight $\leq w$
      then stop and return it.
```

Table 6: The WF for each RM code.

	$(n, k, d_{\min}, w)$	WF
RM(4, 12)	(4096, 794, 256, 1285)	$\leq 2^{-373}$
RM(6, 12)	(4096, 2510, 64, 480)	$\leq 2^{-490}$
RM(6, 13)	(8192, 4096, 128, 1367)	$\leq 2^{-1232}$

S, H, and Q. In addition, square code attack [10] can be applied to RM code-based cryptosystem with insertion. This attack can find the indices of random inserted rows/columns. However by [9], these three attacks can be prevented by applying pucturing methods to the cryptosystem.

8.3.1 Minder-Shokrollahi's attack

One of the major objects of the attack on the McEliece cryptosystems is to find the permutation matrix Q. Let $C = \text{RM}(r, m)^\sigma$ be the permuted code of $\text{RM}(r, m)$ for some unknown permutation σ. In the Minder-Shokrollahi's attack, the attack procedure to find σ is composed of three steps as follows:

1. Find codewords in C, which belong to $\text{RM}(r-1, m)^\sigma$. It is required to find enough number of such codewords to build a basis of $\text{RM}(r-1, m)^\sigma$.
2. Iterate the previous step until obtaining $\text{RM}(1, m)^\sigma$.
3. Determine a permutation τ such that $\text{RM}(1, m)^{\tau\cdot\sigma} = \text{RM}(1, m)$. Then we have $\text{RM}(r, m)^{\tau\cdot\sigma} = \text{RM}(r, m)$. Then τ becomes Q^{-1}.

As you can see, the first step is crucial for the success of this attack. Let $x \in C$ be a minimum weight codeword. Then, we define $C_{\text{supp}(x)}$ as the shortened code of C on $\text{supp}(x)$, that is, find only codewords which are zero on $\text{supp}(x)$ in C and then puncture their components with indices in $\text{supp}(x)$. For example, for $\text{supp}(x) = \{1, 4\}$, we have $c' = (c_2, c_3, c_5, \cdots, c_n) \in$

$C_{\mathrm{supp}(x)}$. Clearly, the length of codewords in $C_{\mathrm{supp}(x)}$ is $n - |\mathrm{supp}(x)|$. Then, it is known that $C_{\mathrm{supp}(x)}$ is a concatenated code defined as

$$C_{\mathrm{supp}(x)} \subseteq \overbrace{\mathrm{RM}(r-1, m-r) \times \mathrm{RM}(r-1, m-r) \times \cdots \times \mathrm{RM}(r-1, m-r)}^{2^r-1} \tag{4}$$

where $\times$ denotes the direct product defined in [7].

Since the permutation is unknown, the position of $\mathrm{RM}(r-1, m-r)$ in $C_{\mathrm{supp}(x)}$ is supposed to be also unknown. However, the algorithm to find the position of $\mathrm{RM}(r-1, m-r)$ is proposed by the Minder-Shokrollahi [7] and thus a codeword in $\mathrm{RM}(r-1, m)^{\sigma}$ can be determined, which corresponds to the first step in the above attack.

8.3.2 Security against Minder-Shokrollahi's attack

Let C be a permutated RM code, i.e., $C = \mathrm{RM}(r, m)^{\sigma}$ by the permutation σ, that is, the permutation matrix Q and x be the minimum weight codeword in C. We would like to find the minimum $|L_D|$, where the Minder-Shokrollahi's attack does not work. Remember that all punctured positions are included in $\mathrm{supp}(x)$ in the proposed puncturing method. Let C' be the punctured code of C by the index set L_D. The first step of the Minder-Shokrollahi's attack is to find the minimum weight codewords. Let x' be a minimum weight codeword of C', which is a punctured codeword of $x \in C$. Then we can find $C'_{\mathrm{supp}(x')}$. The support set of the punctured codeword is denoted as

$$\mathrm{supp}(x') = \{a_1, a_2, \cdots, a_{2^{m-r}-|L_D|}\}$$

with $|\mathrm{supp}(x')| = 2^{m-r} - |L_D|$.

Clearly, the code lengths of $C_{\mathrm{supp}(x)}$ and $C'_{\mathrm{supp}(x')}$ are the same and $C_{\mathrm{supp}(x)} \subseteq C'_{\mathrm{supp}(x')}$ since all deleted positions are included in $\mathrm{supp}(x)$. Now, we are interested in the case of $C_{\mathrm{supp}(x)} \neq C'_{\mathrm{supp}(x')}$, that is, $C_{\mathrm{supp}(x)} \subsetneq C'_{\mathrm{supp}(x')}$, for which the Minder-Shokrollahi's attack does not work. In the following theorem, we can determine the minimum number of punctured positions in order to prevent the Minder-Shokrollahi's attack.

Theorem 3. [9] *For an RM code* $\mathrm{RM}(r, m)$, *at least* $|L_D| = 2^{m-2r}$ *is required for* $C_{\mathrm{supp}(x)} \subsetneq C'_{\mathrm{supp}(x')}$. *And the support set of the minimum weight codeword in* $\mathrm{proj}_{\mathrm{supp}(x)}(C)$ *is the essential punctured locations, where* x *is the minimum weight codeword of* C.

Proof. It is not difficult to check that $\mathrm{RM}(r, m-r) = \mathrm{proj}_{\mathrm{supp}(x)}(C)$. Thus, the minimum weight codeword of $\mathrm{proj}_{\mathrm{supp}(x)}(C)$ is 2^{m-2r}. Let y be the minimum weight codeword in $\mathrm{proj}_{\mathrm{supp}(x)}(C)$. Then, there exists $z \in C$ such that

$$y = \mathrm{proj}_{\mathrm{supp}(x)}(z). \tag{5}$$

Then, $\mathrm{proj}_{N\setminus\mathrm{supp}(x)}(z)$ clearly belongs to $C'_{\mathrm{supp}(x')}$ but not to $C_{\mathrm{supp}(x)}$, where $N = \{1, 2, \cdots, n\}$. Thus, $C_{\mathrm{supp}(x)} \subsetneq C'_{\mathrm{supp}(x')}$. $\square$

Example 1. *Consider an RM code RM$(2,5)$. Suppose that the permutation matrix and scrambling matrix are identity matrices for simplicity. Clearly, one of the minimum weight codewords is* $x = (1111111100\cdots00) \in RM(2,5)$ *and* $\mathrm{proj}_{\mathrm{supp}(x)}(C) = \mathrm{RM}(2,3)$*. Then, one of the minimum weight codewords in* $\mathrm{proj}_{\mathrm{supp}(x)}(C)$ *is* $y = (10001000)$*. Also, we set* $L_D = \{1,5\}$ *and the punctured codeword of* x *is* $x' = (?111?11100\cdots00) = (11111100\cdots00)$*. And* z *in (5) is*

$$z = (10001000|10001000|10001000|10001000). \tag{6}$$

Since $C_{\mathrm{supp}(x)}$ *forms* $\mathrm{RM}(1,3) \times \mathrm{RM}(1,3) \times \mathrm{RM}(1,3)$*,* $\mathrm{proj}_{N\setminus\mathrm{supp}(x)}(z)$ *does not belong to* $C_{\mathrm{supp}(x)}$*. However,* $\mathrm{proj}_{N\setminus\mathrm{supp}(x)}(z)$ *belongs to* $C'_{\mathrm{supp}(x')}$ *by definition. And* $C'_{\mathrm{supp}(x')} \subseteq (\mathrm{RM}(1,3)+\{\mathbf{0},\,(10001000)\})\times(\mathrm{RM}(1,3)+\{\mathbf{0},\,(10001000)\})\times(\mathrm{RM}(1,3)+\{\mathbf{0},\,(10001000)\})$*.*

Example 2. *Consider an RM code, RM$(1,4)$. Suppose that the permutation matrix and scrambling matrix are identity matrices for simplicity. Then, one of the minimun weight codewords is* $x = (1111111100000000)$*. Then,* $\mathrm{proj}_{\mathrm{supp}(x)}(C) = \mathrm{RM}(1,3)$ *and minimum weight of* $\mathrm{proj}_{\mathrm{supp}(x)}(C)$ *is 4. Thus, to prevent the Minder-Shokrollahi's attack, at least 4 components should be punctured. If we puncture less than 4 components, the attack cannot be prevented. This can be described as follows. Let* $L_D = \{1,2,3\}$*, where* $|L_D| = 2^{m-2r} - 1 = 3$*. Let* C' *be a punctured code and* $x' = (111110\cdots0)$*. Then the generator matrix of* $C'_{\mathrm{supp}(x')}$ *is*

$$C'_{\mathrm{supp}(x')} = (0\;0\;0\;0\;0\;0\;0\;0\;1\;1\;1\;1\;1\;1\;1\;1). \tag{7}$$

The $C'_{\mathrm{supp}(x')}$ *is equal to* $C_{\mathrm{supp}(x)}$ *and attacker can proceed to the next step of the Minder-Shokrollahi's attack. But, if* $L_D = \{1,2,3,4\}$*, the generator matrix of* $C'_{\mathrm{supp}(x')}$ *is given as*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{8}$$

Then, we have $C_{\mathrm{supp}(x)} \subsetneq C'_{\mathrm{supp}(x')}$*.*

Similarly, it is not difficult to check that

$$C'_{\mathrm{supp}(x')} \subseteq \overbrace{C'_i \times C'_i \times \cdots \times C'_i}^{2^r-1} = (C'_i)^{2^r-1} \tag{9}$$

where

$$C'_i = \mathrm{RM}(r-1, m-r) + \{\mathbf{0}, \text{an element in the basis of } \mathrm{RM}(r, m-r)\}. \tag{10}$$

It is difficult to correctly decompose component codes C'_i from $C'_{\mathrm{supp}(x')}$ because many random component codes of $C'_{\mathrm{supp}(x')}$ can form $\mathrm{RM}(r-1, m-r) + \{\mathbf{0}, \text{an element in the basis of } \mathrm{RM}(r, m-r)\}$.

8.3.3 Chizhov-Borodin's 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 RM(r, m), can also be constructed in low polynomial-time complexity. Thus, RM$(kr + l(m-1), m)$ can be obtained and finally we have RM$(\gcd(r, m-1), m)$. If $\gcd(r, m-1) = 1$, then RM$(1, m)$ is directly found. Otherwise, RM$(r-1, m)$ can be obtained by the Minder-Shokrollahi's attack. By iterating this procedure until we have $\gcd(r-k, m-1) = 1$, RM$(1, m)$ can be found. Then it is straightforward to find the permutation τ, that is, Q^{-1}.

8.3.4 Security against Chizhov-Borodin's attack

The Chizhov-Borodin's attack uses the property that the dual code of the RM code is also the RM code. For the punctured RM codes, their dual codes are the shortened RM codes [13]. It is not possible to recover the RM codes from the shortened RM codes, because some rows and columns are deleted from the generator matrix. Therefore, the Chizhov-Borodin's attack cannot be applied to the McEliece cryptosystem based on the proposed punctured RM codes.

Example 3. *Let G be a generator matrix of RM$(1,3)$. G_D denotes a generator matrix of a punctured RM code of RM$(1,3)$, where the first and the second columns of the generator matrix are deleted. Then we have*

$$G_D = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{11}$$

and the generator matrix of its dual code is given as

$$G_D{}^{\perp} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{12}$$

The dual code of RM$(1,3)$ is also RM$(1,3)$ but the dual code of the punctured RM code is the shortened code of the RM code. That is, $G_D^{\perp}$ is the generator matrix of the shortened RM code of RM$(1,3)$ but the deleted rows by shortening cannot be recovered.

8.3.5 Square code attack

According to [10], applying insertion of random columns to the McEliece cryptosystem based on RM codes is insecure from the square code attack, which uses a property of the product of the random-column inserted RM codes. The product of codes is defined as follows.

Definition 4 (Product of codes)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be linear codes of length n. Then the product code denoted by $\mathcal{A} * \mathcal{B}$ is the vector space spanned by all the componentwise product $\mathbf{a} \cdot \mathbf{b} = (a_1 b_1, \cdots, a_n b_n)$, where $\mathbf{a} \in \mathcal{A}$ and $\mathbf{b} \in \mathcal{B}$. When $\mathcal{A} = \mathcal{B}$, $\mathcal{A} * \mathcal{A}$ is called the square code of $\mathcal{A}$, denoted by $\mathcal{A}^2$.*

Table 7: Required complexity for operations of the square code attack on $\mathrm{RM}(r, 2r)$ code.

	Number of square code attack cases	Required operations
$\mathrm{RM}(5, 10)$	$2^{17.5}$	$2^{67.5}$
$\mathrm{RM}(6, 12)$	$2^{21.3}$	$2^{81.3}$

Let G_L be a $k \times (n + |L_I|)$ matrix obtained by inserting $|L_I|$ random columns into the generator matrix of an RM code $\mathrm{RM}(r, m)$ and $\mathcal{C}$ be the code spanned by the rows of G_L. The index set $L_I \subset \{1, \cdots, n + |L_I|\}$ is the set of indices that defines inserted locations of random columns. And let $\mathcal{C}_i$ be the code generated by the matrix $G_{L,i}$ obtained by deleting the ith column of G_L. Then the following two cases occur with high probability.

$$\mathrm{dim}\mathcal{C}_i^2 = \begin{cases} \mathrm{dim}\mathcal{C}^2 - 1, & \text{if } i \in L_I \\ \mathrm{dim}\mathcal{C}^2, & \text{if } i \notin L_I. \end{cases} \tag{13}$$

With this argument, an attacker can discover the set L_I using public key of the McEliece cryptosystem based on RM codes in the polynomial time.

8.3.6 Security against square code attack

For $m = 2r$, let us consider an RM code $\mathrm{RM}(r, m)$, with insertion of $|L_I|$ random columns, where the code length is $2^m + |L_I|$. Then, its square code becomes $\mathcal{C}^2 = \mathrm{RM}(m, m)$ with $|L_I|$ random column insertions. The minimum weight codeword of $\mathrm{RM}(m, m)$ is 1 and there are m minimum weight codewords in $\mathrm{RM}(m, m)$. Let x be one of the minimum weight codewords in $\mathrm{RM}(m, m)$. Assume that x has 1 in jth position. If we delete the jth column of G, then $\mathrm{dim}\mathcal{C}_j^2 = \mathrm{dim}\mathcal{C}^2 - 1$ with high probability. Thus there are $|L_I| + m$ indices of reducing dimension of C by 1. Thus, the probability to find randomly inserted columns is $1/\binom{|L_I|+m}{|L_I|}$.

If $m = 10$ and $|L_I| = 10$, $1/\binom{|L_I|+m}{|L_I|} = 1/\binom{20}{10} \approx 2^{-17.5}$. In addition, it requires $O(n^5) \approx 2^{50}$ operations for each case [10]. Thus, inserting 10 random columns requires $2^{67.5}$ more operations to succeed in the square code attack. Since the attacker has to apply the Minder-Shokrollahi's attack after success of the square code attack, it takes more time. The required complexity for operations of the square code attack on $\mathrm{RM}(r, 2r)$ for $r = 5, 6$ is given in Table 7. Therefore, it is difficult to apply the square code attack on $\mathrm{RM}(r, 2r)$ with insertion of random columns.

For $m > 2r$, using the puncturing of the RM codes, we can also prevent the square code attack because (13) does not hold anymore. In the case of $m = 2r$, we can find that if there is a codeword with Hamming weight 1 in C^2, the dimension of C^2 can be reduced by deleting a column. Although the square code of $\mathrm{RM}(2r, m)$ does not contain codeword with Hamming weight 1, we can control the Hamming weight by puncturing method. It is known that the square code is $\mathrm{RM}(2r, m)$ and the minimum Hamming weight of the square code is

2^{m-2r}. Since $2^{m-2r} \geq 2$, deleting one column does not reduce the dimension [10]. However, puncturing more than $2^{m-2r}-1$ columns, (13) does not hold similarly to the case of $\mathrm{RM}(r, 2r)$. Thus the square code attack cannot be applied to the proposed McEliece cryptosystems. The minimum Hamming weight of punctured code in $\mathrm{RM}(r, m)^2$ is reduced when we puncture the codeword in the support set of the minimum Hamming weight codeword in $\mathrm{RM}(r, m)$. Also, the square code attack is effectively prevented when $m-2r = 1, 2$, because the number of required puncturing bits is small.

8.4 EUF-CMA security

The proposed pqsigRM is designed for EUF-CMA security. By inserting random matrix R, it is hard to distinguish H' to random parity check matrix H_R. See Appendix A for the proof of EUF-CMA security.

9 Advantages and limitations (2.B.6)

The advantages of pqsigRM signature scheme are the controllable signing time and EUF-CMA secure. Comparing to the CFS signature scheme, the signing time does not depend on error correction capability t. Also the signing time and security level are controllable by the parameter setting. The trade off between signing time and security level are controlled by parameters N and w. Also, by inserting random matrix R, it is hard to distinguish H' to random parity check matrix H_R. Thus, the pqsigRM satisfies EUF-CMA security.

The limitation of pqsigRM is the relatively large public key size. Because the RM code is not quasi cyclic, the key size of public key is $(n-k) \times k$.

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] R. Overbeck and N. Sendrier, "Code-based cryptography," *Post-Quantum Cryptography,* pp. 95–146, Springer.

[6] J. Stern, “A method for finding codewords of small weight,” *Coding Theory and Applications*, vol. 388, pp. 106–133, 1989

[7] L. Minder and A. Shokrollahi, “Cryptanalysis of the Sidelnikov cryptosystem,” in *Proc EUROCRYPT* 2007, LNCS, vol. 4515, 2007, pp. 347–360.

[8] 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).

[9] W. Lee, J.-S. No, and Y.-S. Kim, “Punctured Reed–Muller code-based McEliece cryptosystems,” *IET Commun.*, vol. 11, no. 10, pp. 1543–1548, Jul. 2017.

[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] W. Lee, Y.-S. Kim, and J.-S. No, “A new signature scheme based on punctured Reed-Muller code with random insertion,” arXiv 1711.00159v1, Nov. 2017

[13] E. C. Boyle and R.J. McEliece, “Asymptotic weight enumerators of randomly punctured, expurgated, and shortened code ensembles.” in *Proc. 46th Annual Allerton Conf. Commun., Control, and Computing*, Sep. 2008, pp. 910–917

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

[15] 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.

A Appendix: Proof of EUF-CMA security

The proposed signature scheme can be simplified as follows. We can consider finding e satisfying $H'e^T = h(h(M)|i)$ as a signing process, where the parity check matrix of a linear code is H_m, and γ is the decoding algorithm in the signature scheme in the previous section. Then, the proposed signature scheme can be considered to be the same as the original CFS scheme for the EUF-CMA security check.

1) Key Generation:

Let H_m be a parity check matrix of a modified RM code with the decoding algorithm γ. Then, the private keys are S, Q, and H_m, and the public keys are $H' = SH_mQ$ and w.

2) Signing:

```
To sign a message $M$,
For $i = 1$ to $N$
    $e' = \gamma(S^{-1}h(h(M)|i))$
    if $\mathrm{wt}(e') \le w$, go to *.
end
* Output $(M, e = e'(Q^{-1})^T, i)$
```

3) Verification

Check $\mathrm{wt}(e) \le w$ and $H'e^T = h(h(M)|i)$. If TRUE, then return ACCEPT; otherwise, return REJECT.

To prove the EUF-CMA security, we need the following assumption and proposition. The differences between the proposed signature scheme and the CFS signature scheme are; i) the use of a different code, namely, a modified RM code rather than a Goppa code, and ii) the use of complete decoding instead of syndrome decoding.

Assumption 5 (RM code distinguishability problem)**.** *There is no probabilistic polynomial time (PPT) distinguisher $\mathcal{D}$ that can distinguish $H' = SH_mQ$ from a randomly generated parity check matrix H_R.*

To the best of our knowledge, there are no known algorithms for distinguishing a modified parity check matrix H' of a modified RM code from H_R up to now, and thus we set the following assumption.

Proposition 6 ([15] Hardness of complete decoding)**.** *The complete decoding problem for an (n, k, t) linear code is an NP-complete problem if the Hamming weight of e is less than $n/3$.*

With this assumption and proposition, the following theorem holds.

Theorem 7. *The proposed modified RM code-based signature scheme is EUF-CMA secure.*

Proof. The proof of this theorem is almost the same as the proof of the EUF-CMA security of the CFS signature scheme [14]. It was proved in [14] that a variant of a CFS scheme is EUF-CMA secure if certain assumptions are true. However, it was shown that the assumption that distinguishing Goppa codes from random codes is difficult is not true for the case of some parameters (small t) used in the CFS signature scheme. Thus, we can follow the logic of the proof in [14] because the adopted assumption for the proposed modified RM code-based signature scheme is still valid. We define the sequence of games $G_0, G_1, \cdots, G_5$ in the same way as in [14]. Let G_0 be the original security game, that is, the EUF-CMA game, and G_5 be solving the syndrome decoding problem.

The main differences between the proposed signature scheme and the CFS signature scheme are mostly in Games G_3 and G_5. In the proof of the CFS signature scheme, Game G_3 discusses the Goppa code distinguishing problem, but for a small t, it turns out to be distinguishable from a random code. In the case of the proposed signature scheme, we adopt the modified RM code distinguishing problem in Assumption 5 because there has been no way to prove the distinguishability up to now. Although Game G_5 is related to the syndrome decoding problem in the proof of the original CFS signature scheme, we will replace it using the complete decoding problem, which is known as an NP-complete problem. A full description of this proof is given as follows.

The challenger $\mathcal{C}$ plays a sequence of games $G_0, G_1, \cdots, G_5$. Here, G_0 corresponds to the standard EUF-CMA game as mentioned above. In G_0, the adversary $\mathcal{A}$ tries to forge a signature. If the adversary $\mathcal{A}$ successfully forges the signature, then $\mathcal{A}$ wins the game G_0. Successive games are given through slight modifications of the preceding games. Let $\Pr[G_i]$ be the winning probability of each game G_i. We then have to prove that the probability of the winning condition of these games is proved to be arbitrarily small through all of the intermediate games.

G_0: The challenger $\mathcal{C}$ obtains the private and public keys using a key generation algorithm. The adversary $\mathcal{A}$ obtains the public key H', and can access a hash oracle $\mathcal{H}$ and signing oracle Σ. Let q_h and q_s be the maximum numbers of queries made by the adversary $\mathcal{A}$ to the hash oracle and the signing oracle, respectively. The procedure of G_0 is given in Algorithm 5. Then, the winning probability of G_0 is given as

$$\Pr[G_0] = \mathrm{succ}^{\mathrm{EUF-CMA}}(\mathcal{A}). \tag{14}$$

Algorithm 5 G_0 (EUF-CMA)

```
1. $(H', S, H, Q) \leftarrow \mathrm{keygen}(\mathcal{C})$
2. Set the oracles $\mathcal{H}$ and $\Sigma$
3. $(M^*, \sigma^*, i^*) \leftarrow \mathcal{A}^{\mathcal{H},\Sigma}(H')$
4. If $\mathcal{H}(M^*, i^*) = H'\sigma^{*T}, \mathrm{wt}(\sigma^*) \leq w$, and $\Sigma$ did not provide $\sigma^*$, then
       $\mathcal{A}$ wins the game
   else
       $\mathcal{A}$ loses the game
   end
```

G_1: In this game, the challenger modifies the hash oracle $\mathcal{H}$ by $\mathcal{H}'$. In $\mathcal{H}'$, the challenger uses a list Λ that consists of counter values of $i = \Lambda(M)$ for message M such that $\mathcal{H}(M, i)$ is a decodable syndrome and another list $\Lambda_{\mathcal{H}}$ to store a valid syndrome-error pair that was already produced in the previous queries. If there is no element corresponding to the input, the output is $\perp$. The modified hash oracle $\mathcal{H}'$ produces syndromes according to Algorithm 6, and finally produces $q_h + q_s + 1$ syndromes. In addition, it is known that the relation of $\Pr[G_0]$ and $\Pr[G_1]$ is given as

$$|\Pr[G_1] - \Pr[G_0]| \le \epsilon_0 \tag{15}$$

where $\epsilon_0 = 1 - \left(1 - \frac{1}{2^{n-k}}\right)^{q_h+q_s+1}$ [14].

Algorithm 6 Game G_1 ($\mathcal{H}'$: simulation of $\mathcal{H}$)

```
Input: a pair $(M, i)$
Output: a syndrome $s$

1. If $\Lambda(M) = \perp$, then
   $\Lambda(M) \xleftarrow{R} \{1, \cdots, 2^{n-k}\}$

2. $(s, e) \leftarrow \Lambda_{\mathcal{H}}(M, i)$

3. If $i \ne \Lambda(M)$, then

      If $s = \perp$, then
         $s \xleftarrow{R} F_2^{n-k}$
         $\Lambda_{\mathcal{H}}(M, i) \leftarrow (s, \perp)$
      end
      return $\mathcal{H}(M, i) = s$

   else

      If $s = \perp$, then
         $e \xleftarrow{R} \{y \in F_2^n | \mathrm{wt}(y) \le w\}$
         $s \leftarrow He^T$
         $\Lambda_{\mathcal{H}}(M, i) \leftarrow (s, e)$
      end
      return $\mathcal{H}'(M, i) = s$

   end
```

G_2: In G_2, the challenger replaces the signing oracle Σ with Σ'. The modified signing oracle queries $\mathcal{H}'$ on $(M, \Lambda(M))$ according to Algorithm 7. In addition, the winning probability relation of G_1 and G_2 is derived as

$$|\Pr[G_2] - \Pr[G_1]| \le \epsilon_1 \tag{16}$$

where $\epsilon_1 = 1 - \left(1 - \frac{q_s}{2^{n-k}}\right)^{q_h}$.

Algorithm 7 Game G_2 (Σ': simulation of Σ)

Input: a message M
Output: a signature (i, σ)

```
1. If $\Lambda(M) = \perp$, then
       $\Lambda(M) \overset{R}{\leftarrow} \{1, \cdots, 2^{n-k}\}$
   end
2. $\mathcal{H}'(M, \Lambda(M))$
3. $(s, x) \leftarrow \Lambda_{\mathcal{H}'}(M, \Lambda(M))$
4. $\Lambda(M) \leftarrow \perp$
5. Return $\Sigma(M) = (i, x)$
```

G_3: In G_3, the challenger replaces the key generation algorithm with the selection of a random binary parity check matrix. The selected parity check matrix is taken as the public key. Because neither the hash oracle nor the signature oracle uses the hash function and the private keys, the difference in the winning probabilities of G_2 and G_3 is the same as the distinguishing probability between the modified RM code and a random binary code, that is,

$$|\Pr[G_3] - \Pr[G_2]| \leq \epsilon_{\text{distinguish}}. \tag{17}$$

By Assumption 5, the value of $\epsilon_{\text{distinguish}}$ is negligible. The description of G_3 is given as Algorithm 8.

Algorithm 8 Game G_3

Input: a parity check matrix H
Output: a bit b

```
1. Given $w$, set the oracles $\mathcal{H}'$ and $\Sigma'$
2. $(M^*, \sigma^*, i^*) \leftarrow \mathcal{A}^{\mathcal{H}', \Sigma'}(H)$
3. If $\mathcal{H}'(M^*, i^*) = H\sigma^{*T}, \mathrm{wt}(\sigma^*) \leq w$, and $\Sigma'$ did not provide $\sigma^*$, then
       $b = 1$
   else
       $b = 0$
   end
```

G_4: G_4 is conditioned by an adversary making a forgery on a particular hash query. The challenger first obtains a random $c \overset{R}{\leftarrow} \{1, \cdots, q_h + q_s + 1\}$. Adversary $\mathcal{A}$ wins the game if the cth query to $\mathcal{H}'$ is made on (M^*, i^*). Because c is randomly chosen from $q_h + q_s + 1$ possibilities, the winning probability of G_4 is given as

$$\Pr[G_4] = \frac{\Pr[G_3]}{q_h + q_s + 1}. \tag{18}$$

G_5: In this game, the challenger modifies the hash oracle to output a random syndrome s^* to the cth query. The winning probability of G_5 is the same as the winning probability of G_4. The detailed procedure for G_5 is given in Algorithm 9. Note that this game is the same as solving the complete decoding problem. Then,

$$\Pr[G_5] = \Pr[G_4] \leq \epsilon_{\text{complete}}. \tag{19}$$

From Proposition 6, the value of $\epsilon_{\text{complete}}$ is negligible.

Algorithm 9 Game G_5

```
Input: an adversary $\mathcal{A}$
  1. $c \overset{R}{\leftarrow} \{1, \cdots, q_{\mathcal{H}} + q_{\Sigma} + 1\}$
  2. $H^* \overset{R}{\leftarrow} (n, k)$ binary code for given $w$
  3. $s^* \overset{R}{\leftarrow} F_2^{n-k}$
  4. Set the oracles $\mathcal{H}'$ and $\Sigma'$
  5. $(M^*, \sigma^*, i^*) \leftarrow \mathcal{A}^{\mathcal{H}', \Sigma'}(H^*)$
  6. If $\begin{cases} \mathcal{H}'(M^*, i^*) = H\sigma^{*T} \\ \text{wt}(\sigma^*) \leq w \end{cases}$ and $\begin{cases} \Sigma' \text{ did not provide } \sigma^* \\ c-\text{th query to } \mathcal{H}' \text{ was } (M^*, i^*), \end{cases}$
     then
        $\mathcal{A}$ wins the game
     else
        $\mathcal{A}$ loses the game
     end
```

Combining all of the above equations, (14)–(19), we have

$$\text{succ}^{\text{EUF-CMA}}(\mathcal{A}) \leq (q_h + q_s + 1)\epsilon_{\text{complete}} + \epsilon_{\text{distinguish}} + \epsilon_0 + \epsilon_1. \tag{20}$$

Hence, the probability of a successful forgery is negligible if the punctured RM codes with random insertions are indistinguishable from random linear codes and the complete decoding problem is intractable. Thus, the proposed signature scheme is EUF-CMA secure. □

B Statements

These statements “must be mailed to Dustin Moody, Information Technology Laboratory, Attention: Post-Quantum Cryptographic Algorithm Submissions, 100 Bureau Drive – Stop 8930, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, or can be given to NIST at the first PQC Standardization Conference (see Section 5.C).”

First blank in submitter statement: full name. Second blank: full postal address. Third, fourth, and fifth blanks: name of cryptosystem. Sixth and seventh blanks: describe and enumerate or state “none” if applicable.

First blank in patent statement: full name. Second blank: full postal address. Third blank: enumerate. Fourth blank: name of cryptosystem.

First blank in implementor statement: full name. Second blank: full postal address. Third blank: full name of the owner.

B.1 Statement by Each Submitter

I, _________________________, of ____, do hereby declare that the cryptosystem, reference implementation, or optimized implementations that I have submitted, known as _________________________, is my own original work, or if submitted jointly with others, is the original work of the joint submitters. I further declare that (check one):

- *I do not hold and do not intend to hold any patent or patent application with a claim which may cover the cryptosystem, reference implementation, or optimized implementations that I have submitted, known as _________________________ OR (check one or both of the following):*
  - *to the best of my knowledge, the practice of the cryptosystem, reference implementation, or optimized implementations that I have submitted, known as _________________ may be covered by the following U.S. and/or foreign patents:*

    ____
  - *I do hereby declare that, to the best of my knowledge, the following pending U.S. and/or foreign patent applications may cover the practice of my submitted cryptosystem, reference implementation or optimized implementations:*

    ____

I do hereby acknowledge and agree that my submitted cryptosystem will be provided to the public for review and will be evaluated by NIST, and that it might not be selected for standardization by NIST. I further acknowledge that I will not receive financial or other compensation from the U.S. Government for my submission. I certify that, to the best of my knowledge, I have fully disclosed all patents and patent applications which may cover my cryptosystem, reference implementation or optimized implementations. I also acknowledge and agree that the U.S. Government may, during the public review and the evaluation process, and, if my submitted cryptosystem is selected for standardization, during the lifetime of the standard, modify my submitted cryptosystem's specifications (e.g., to protect against a newly discovered vulnerability).

I acknowledge that NIST will announce any selected cryptosystem(s) and proceed to publish the draft standards for public comment.

I do hereby agree to provide the statements required by Sections 2.D.2 and 2.D.3, below, for any patent or patent application identified to cover the practice of my cryptosystem, reference implementation or optimized implementations and the right to use such implementations for the purposes of the public review and evaluation process.

I acknowledge that, during the post-quantum algorithm evaluation process, NIST may remove my cryptosystem from consideration for standardization. If my cryptosystem (or the derived cryptosystem) is removed from consideration for standardization or withdrawn from consideration by all submitter(s) and owner(s), I understand that rights granted and assurances made under Sections 2.D.1, 2.D.2 and 2.D.3, including use rights of the reference and optimized implementations, may be withdrawn by the submitter(s) and owner(s), as appropriate.

Signed:

Title:

Date:

Place:

B.2 Statement by Patent (and Patent Application) Owner(s)

If there are any patents (or patent applications) identified by the submitter, including those held by the submitter, the following statement must be signed by each and every owner, or each owner's authorized representative, of each patent and patent application identified.

I, __________________________, of ____, am the owner or authorized representative of the owner (print full name, if different than the signer) of the following patent(s) and/or patent application(s): ________________________________

____

and do hereby commit and agree to grant to any interested party on a worldwide basis, if the cryptosystem known as __________________ is selected for standardization, in consideration of its evaluation and selection by NIST, a non-exclusive license for the purpose of implementing the standard (check one):

- *without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination, OR*
- *under reasonable terms and conditions that are demonstrably free of any unfair discrimination.*

I further do hereby commit and agree to license such party on the same basis with respect to any other patent application or patent hereafter granted to me, or owned or controlled by me, that is or may be necessary for the purpose of implementing the standard.

I further do hereby commit and agree that I will include, in any documents transferring ownership of each patent and patent application, provisions to ensure that the commitments and assurances made by me are binding on the transferee and any future transferee.

I further do hereby commit and agree that these commitments and assurances are intended by me to be binding on successors-in-interest of each patent and patent application, regardless of whether such provisions are included in the relevant transfer documents.

I further do hereby grant to the U.S. Government, during the public review and the evaluation process, and during the lifetime of the standard, a nonexclusive, nontransferrable, irrevocable, paid-up worldwide license solely for the purpose of modifying my submitted cryptosystem's specifications (e.g., to protect against a newly discovered vulnerability) for incorporation into the standard.

Signed:

Title:

Date:

Place:

B.3 Statement by Reference/Optimized Implementations' Owner(s)

The following must also be included:

I, __________________________, ____, am the owner or authorized representative of the owner __________________________ of the submitted reference implementation and optimized implementations and hereby grant the U.S. Government and any interested party the right to reproduce, prepare derivative works based upon, distribute copies of, and display such implementations for the purposes of the post-quantum algorithm public review and evaluation process, and implementation if the corresponding cryptosystem is selected for standardization and as a standard, notwithstanding that the implementations may be copyrighted or copyrightable.

Signed:

Title:

Date:

Place: