IET Communications

Research Article

IET Journals
The Institution of Engineering and Technology

Punctured Reed–Muller code-based McEliece cryptosystems

ISSN 1751-8628
Received on 25th October 2016
Revised 15th March 2017
Accepted on 19th April 2017
E-First on 27th July 2017
doi: 10.1049/iet-com.2016.1268
www.ietdl.org

Wijik Lee[1], *Jong-Seon No*[1], *Young-Sik Kim*[2] ✉

[1]*Department of Electrical and Computer Engineering, INMC, Seoul National University, Seoul 08826, Korea*
[2]*Department of Information and Communication Engineering, Chosun University, Gwangju 61452, Korea*
✉ *E-mail: iamyskim@chosun.ac.kr*

Abstract: The authors propose new McEliece cryptosystems based on punctured Reed–Muller (RM) codes. They successfully show that the commonly known attacks, such as the Minder–Shokrollahi attack, the Chizhov–Borodin attack, and the square code attack, are ineffective against the proposed RM code-based McEliece cryptosystem. We developed an optimal puncturing scheme to prevent the above-mentioned attacks for the proposed RM code-based cryptosystems in a sense that the exact locations of puncturing positions with the minimum number of punctured columns of the generator matrix should be found for attacking. It is important to carry out the minimum number of punctures, however, as code modification resulting from puncturing can reduce security. Additionally, the square code attack can be prevented in the proposed RM code-based McEliece cryptosystems by using both the proposed puncturing method and random insertion methods.

1 Introduction

It is widely known that most conventional public key cryptosystems, such as RSA cryptosystem, elliptic curve cryptosystem, and others, can be broken by using sophisticated operations over quantum computers. Thus, many studies have been devoted to developing robust cryptosystems resistant to attacks by quantum computers; these are known as post-quantum cryptosystems. One well-known post-quantum cryptosystem is a code-based cryptosystem using Goppa code, which was first introduced by McEliece [1]. Although encryption and decryption of the McEliece cryptosystem are usually faster than those of the conventional cryptosystems such as RSA and elliptic curve cryptosystems, it requires very large public and private key sizes. Thus, there have been many efforts to reduce the key sizes of the McEliece cryptosystems.

One approach to reducing key sizes is adopting other error correcting codes in place of the Goppa code [2–4] and utilising their mathematical structures. For example, the generalised Reed–Solomon (GRS) code [2], the polar code [3], and the Reed–Muller (RM) code [4] have been used for the code-based cryptosystems. In the GRS code-based cryptosystem, the private key matrix is determined by two vectors $\boldsymbol{\alpha}, \boldsymbol{v}$, and thus the key size of the McEliece cryptosystem can be dramatically reduced. Further, it is known that McEliece cryptosystems using RM codes can add much larger number of errors than the minimum distance of the RM codes, meaning the matrix size can also be reduced while maintaining security levels [4].

However, the McEliece cryptosystem based on RM codes is proved to be insecure due to the Minder–Shokrollahi [5] and the Chizhov–Borodin's attacks. While the structure of error correcting codes helps us to reduce key size, it may also reveal information on the private key to attackers. To avoid this, a McEliece cryptosystem based on the GRS or RM codes with random-column insertion for the generator matrix is proposed [6, 7]. It was found; however, it turns out that the proposed system can be broken by square code attacks [8].

In this paper, we propose a modification of McEliece cryptosystems based on punctured RM codes. In this modification, some columns of the generator matrix of the original RM codes are carefully punctured to prevent effective cryptanalysis. In fact, the puncturing of the generator matrix in the McEliece cryptosystem was considered in the quasi-cyclic-low-density parity check (QC-LDPC) code-based McEliece cryptosystem [9]. However, in [9], only the security of the QC-LDPC code-based cryptosystem in terms of information set decoding was analysed. Alternatively, we focus on the effect of puncturing of the generator matrix to determine how many columns should be punctured and which locations are most effective to hide the mathematical structure of the code from attackers.

Here, we focus on the McEliece cryptosystem based on RM codes, while the modification of the RM codes by puncturing can be applied to the other code-based McEliece cryptosystems. We will show that if public keys are properly modified by the proposed sophisticated puncturing of the generator matrix, with or without randomly inserting random columns into the punctured generator matrix, all known attacks for the McEliece cryptosystem based on RM codes are successfully repelled. While an attack with a randomly permuted and scrambled generator matrix is not able to find the mathematical structure of the original code, the legitimate receiver can reconstruct the original message because they know the exact locations of punctured and inserted columns of the generator matrix. We perform security analysis with respect to the known attacks for the proposed McEliece cryptosystem based on punctured RM codes with successful results.

This paper is organised as follows: In Section 2, we present the preliminaries for the McEliece cryptosystems based on the RM codes and their attack algorithms. In Section 3, we propose a modified McEliece cryptosystems based on the RM code with sophisticated column puncturing of the generator matrix with or without inserting random columns and verify the its security against the various known attacks in Section 4. Section 5 presents our conclusions.

2 Preliminaries

In this section, we introduce RM codes, the McEliece cryptosystem, and various attack algorithms for the McEliece cryptosystem.

2.1 RM codes

The RM code RM(r,m) is a linear code defined by Boolean functions of m variables and degrees less than or equal to r for any integers m and r with $0 \leq r \leq m$. Boolean functions of m variables are evaluated on 2^m different positions, which correspond to a code

word with a length of 2^m in RM(r,m). RM(r,m) is the set of code words obtained by evaluating all Boolean functions of m variables and degrees less than or equal to r. The set of Boolean functions in the variables $v_1, \ldots, v_m$ of degrees less than or equal to r is denoted by $B(r, \{v_1, \ldots, v_m\})$. The fact that all Boolean functions generating code words in $B(r-1, \{v_1, \ldots, v_m\})$ are also in $B(r, \{v_1, \ldots, v_m\})$ implies the following proposition:

Proposition 1 (Minder and Shokrollahi, 2007 [4]): For any integer m, we have

$$\mathrm{RM}(0, m) \subset \mathrm{RM}(1, m) \subset \cdots \subset \mathrm{RM}(m, m).$$

The code length n, dimension k, and the minimum distance d of RM(r,m) are given as

$$n = 2^m, \quad k = \sum_{i=0}^{r} \binom{m}{i}, \quad d = 2^{m-r}.$$

We require the following definitions for the proposed RM code-based McEliece cryptosystem:

Definition 1: The support of a codeword $\boldsymbol{c} \in \mathrm{RM}(r,m)$ is defined as the set of indices i such that $c_i \neq 0$, which is denoted by $\mathrm{supp}(c)$.

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

Example 1: Let $\boldsymbol{c} = (11011001)$ and $L = \{1, 2, 3, 7\}$. Then, $\mathrm{proj}_L(\boldsymbol{c}) = (1100)$, which is composed of the first, second, third, and seventh components of $\boldsymbol{c}$.

Proposition 2 (Minder and Shokrollahi [5]): Let $\boldsymbol{x}$ be a codeword with the minimum weight in RM(r,m). Then, there exist $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r \in \mathrm{RM}(1, m)$ such that

$$\boldsymbol{x} = \boldsymbol{x}_1 \cdot \boldsymbol{x}_2 \cdot \cdots \cdot \boldsymbol{x}_r$$

where $\boldsymbol{x}_i$ is a codeword with the minimum weight in $\mathrm{RM}(1, m)$ and $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ denotes the componentwise multiplication.

Propositions 1 and 2 are used as the main tools for the Minder–Shokrollahi attack, which will be explained in the following sections.

2.2 McEliece cryptosystems

McEliece introduced a public key cryptosystem based on the difficulty of decoding random linear codes, which consists of three algorithms, key generation, encryption, and decryption, as follows [1]:

Key generation: Let $\boldsymbol{G}$ be a $k \times n$ generator matrix of the (n,k) linear code. Let $\boldsymbol{S}$ be a $k \times k$ scrambling matrix and $\boldsymbol{P}$ an $n \times n$ permutation matrix. Bob generates the public key by calculating $\boldsymbol{G}' = \boldsymbol{SGP}$, where $\boldsymbol{S}$, $\boldsymbol{P}$, and $\boldsymbol{G}$ are the private keys of Bob. The error correction capability t of the linear code with generator matrix $\boldsymbol{G}$ is also disclosed.

Encryption: Alice generates a code word corresponding to the message $m \in \{0,1\}^k$ using Bob's public key $(\boldsymbol{G}', t)$. She chooses the random error vector $\boldsymbol{e} \in \{0,1\}^n$ with a maximum Hamming weight of t and sends the ciphertext $\boldsymbol{c} = \boldsymbol{mG}' + \boldsymbol{e}$ to Bob.

Decryption: When Bob receives the ciphertext $\boldsymbol{c}$, he first multiplies $\boldsymbol{P}^{-1}$ to the right hand side of the ciphertext as $\boldsymbol{cP}^{-1} = \boldsymbol{mSG} + \boldsymbol{eP}^{-1}$. Using a decoding algorithm, Bob finds $\boldsymbol{m}\,\boldsymbol{S}$, and by multiplying $\boldsymbol{S}^{-1}$ he can recover the original message m.

Originally, the generator matrix $\boldsymbol{G}$ of the McEliece cryptosystem is a generator matrix of the Goppa code, and there have subsequently been many suggestions regarding generator matrices of the different linear codes, including RM codes.

In this paper, we focus on the generator matrix of the RM codes. RM code-based cryptosystems were first proposed by Sidelnikov [4]; now referred to as the 'Sidelnikov cryptosystem'. In this cryptosystem, Sidelnikov introduced a number of errors greater than the error correction capability t (in the case of RM code, $2^{m-r-1} - 1$), e.g. for $(n, k, r) = (1024, 176, 3)$ and (2048, 232, 3), the number of errors is > 200 and 400, respectively. Even with these excessive errors, the legitimate receiver can successfully remove them with high accuracy by using an efficient decoding algorithm of the RM code proposed by Sidelnikov and Pershakov [10]. This means that an attacker should correct a larger number of errors than the error correctability t in the ciphertext using decoding of random linear code, which imposes more difficulties on the attacker. However, it was shown that the mathematical structure of RM code in the public key (a randomised generator matrix) reveals the secret information, random permutation matrix, and private key using sophisticated attacking algorithms [5, 11]. It should be noted that the McEliece cryptosystem based on RM codes was broken by three known attacks.

To avoid the Minder–Shokrollahi and Chizhov–Borodin attacks, a modification scheme for generator matrices by inserting random columns into the random positions of the generator matrices was proposed [6–8]. In the subsequent discussion, we will use the following notations for the insertion of random columns into the generator matrix: Let L_1 be a set of inserted column indices of the $k \times (n + |L_1|)$ generator matrix $\boldsymbol{G}_I$. The permutation matrix $\boldsymbol{P}_I$ should be an $(n + |L_1|) \times (n + |L_1|)$ matrix. The encryption procedure is exactly the same as that of the original McEliece cryptosystem. In the decryption procedure, after multiplying P_I^{-1} to the received ciphertext, Bob can delete the elements with indices in L_1 from $m\boldsymbol{SG}_I + eP_I^{-1}$, and the remaining decryption procedure is the same as that of the McEliece cryptosystem. It was shown that when the insertion is solely used for modification, the system is not secure by the square code attacks.

However, by using the random insertion after the sophisticated puncturing of generator matrix of the RM codes, it can be shown that the modified generator matrices do not reveal secret information to attacker. Furthermore, we will show that the RM code-based cryptosystem can be resurrected by using the proposed modification of McEliece cryptosystem based on the punctured RM codes.

2.3 Attacks on McEliece cryptosystems

In this section, we briefly describe the main ideas of decisional cryptanalyses for the McEliece cryptosystem based on RM codes.

2.3.1 Minder–Shokrollahi's attack [4].: One of the major objectives of an attack on the McEliece cryptosystems is to find the permutation matrix $\boldsymbol{P}$. Let $C = \mathrm{RM}(r, m)^\sigma$ be the permuted code of RM(r,m) for an unknown permutation σ. In the Minder–Shokrollahi attack, the attack procedure to find σ is composed of three steps:

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

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

$C_{\mathrm{supp}(\boldsymbol{x})}$ is a concatenated code defined as (see (1)) where $\times$ denotes the direct product defined in [5].

As the permutation is unknown, the position of $\mathrm{RM}(r-1, m-r)$ in $C_{\mathrm{supp}(\boldsymbol{x})}$ is expected to be also unknown. However, the algorithm to find the position of $\mathrm{RM}(r-1, m-r)$ is proposed by Minder and Shokrollahi [5], and thus a code word in $\mathrm{RM}(r-1, m)^{\sigma}$ can be determined, which corresponds to the first step in the above attack.

2.3.2 Chizhov–Borodin's attack [11].: From the 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. Furthermore, $\mathrm{RM}(m-r-1, m)$, a dual code of $\mathrm{RM}(r,m)$, can also be constructed with 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 a straightforward process to find the permutation τ, that is P^{-1}.

2.3.3 Square code attacks [8]: According to [8], applying the insertion of random columns to the McEliece cryptosystem based on RM codes does not protect against square code attacks, which use a property of the product of random-column inserted RM codes. The product of codes is defined as follows:

Definition 3 (product of codes): Let $\mathscr{A}$ and $\mathscr{B}$ be linear codes of length n. The product code denoted by $\mathscr{A} * \mathscr{B}$ as the vector space spanned by all componentwise products $\boldsymbol{a} \cdot \boldsymbol{b} = (a_1 b_1, \ldots, a_n b_n)$, where $\boldsymbol{a} \in \mathscr{A}$ and $\boldsymbol{b} \in \mathscr{B}$. When $\mathscr{A} = \mathscr{B}$, $\mathscr{A} * \mathscr{A}$ is called the square code of $\mathscr{A}$, denoted by $\mathscr{A}^2$.

Let $\boldsymbol{G}_L$ be a $k \times (n + |L_1|)$ matrix obtained by inserting $|L_1|$ random columns into the generator matrix of the RM code $\mathrm{RM}(r,m)$ and $\mathscr{C}$ be the code spanned by the rows of $\boldsymbol{G}_L$. The index set $L_1 \subset \{1, \ldots, n + |L_1|\}$ is the set of indices that defines inserted locations of random columns. And let $\mathscr{C}_i$ be the code generated by the matrix $\boldsymbol{G}_{L,i}$, obtained by deleting the ith column of $\boldsymbol{G}_L$. The following two cases then occur with high probability:

$$\dim \mathscr{C}_i^2 = \begin{cases} \dim \mathscr{C}^2 - 1 & \text{if } i \in L_1 \\ \dim \mathscr{C}^2 & \text{if } i \notin L_1. \end{cases} \tag{2}$$

With this argument, an attacker can discover the set L_1 using the public key of the McEliece cryptosystem based on RM codes in the polynomial time.

3 Modifications of RM code-based cryptosystem

In this section, we propose modifications to the McEliece cryptosystem based on RM code. The proposed cryptosystem starts with the modification of the generator matrix of the RM code $\mathrm{RM}(r,m)$, where $n = 2^m$ and $k = \sum_{i=0}^{r} \binom{m}{i}$. Here, we consider two modifications: (i) modification by puncturing at the minimum number of the specified locations of the generator matrix and (ii) modification by both the sophisticated puncturing and random insertion of columns in the generator matrix.

3.1 Modification by puncturing

The proposed modification of the McEliece cryptosystem can be presented by the following three algorithms:

(1) Key generation

(1-1) *Puncturing:* Let $\boldsymbol{G}$ be a $k \times n$ generator matrix of RM code, $C = \mathrm{RM}(r, m)$. Next find a minimum weight code word $\boldsymbol{x}$ in C and $\mathrm{supp}(\boldsymbol{x})$ and find a minimum weight code word $\boldsymbol{y}$ in $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$. Then, we have the set of indices L_D corresponding to $\mathrm{supp}(\boldsymbol{y})$ in the original code indices, and finally we delete columns with indices in L_D from $\boldsymbol{G}$, which is denoted by $\boldsymbol{G}_D$. This process is presented in Algorithm 1

(1-2) *Generating $\boldsymbol{S}$ and $\boldsymbol{P}$:* Let $\boldsymbol{S}$ be a $k \times k$ scrambling matrix and $\boldsymbol{P}$ be an $(n - |L_D|) \times (n - |L_D|)$ permutation matrix. The public key is generated by calculating $\boldsymbol{G}'_D = \boldsymbol{S}\boldsymbol{G}_D\boldsymbol{P}$. The number of arbitrary random errors $t' = \lfloor t - |L_D|/2 \rfloor$ of $\boldsymbol{G}_D$ is known to others together with $\boldsymbol{G}'_D$ as the public keys. Note that the parameter t is determined according to the Sidelnikov decoding algorithm, which can correct almost all errors to a limit greater than the original error correction capability of RM codes [10]. The private keys are $\boldsymbol{P}$, $\boldsymbol{S}$, $\boldsymbol{G}$, and L_D.

(2) Encryption Alice encrypts a message $m \in \{0,1\}^k$ using Bob's public key $(\boldsymbol{G}_D, t')$. She chooses a random error vector $\boldsymbol{e} \in \{0,1\}^{n - |L_D|}$ with a maximum Hamming weight of t' and sends the ciphertext $\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G}_{\boldsymbol{D}'} + \boldsymbol{e}$ to Bob.

(3) Decryption When Bob receives the ciphertext $\boldsymbol{c}$, he first multiplies $\boldsymbol{P}^{-1}$ to the right hand side of the ciphertext as $\boldsymbol{c}\boldsymbol{P}^{-1} = \boldsymbol{m}\boldsymbol{S}\boldsymbol{G}_D + \boldsymbol{e}\boldsymbol{P}^{-1}$, and then inserts the erasure mark '?' in the jth positions $j \in L_D$ for erasure decoding. Alternatively, Bob can randomly insert '0' or '1' instead of the erasure mark '?' and then apply a conventional decoding algorithm, as the erasures can be treated as errors. After decoding, multiplying $\boldsymbol{S}^{-1}$ to $\boldsymbol{m}\boldsymbol{S}$ ensures he can recover the original message m.

In the modified McEliece cryptosystem, we can exchange the order of encryption procedures, specifically, the puncturing step and key generation step. It is not difficult to check that in the proposed modification of McEliece cryptosystem, deleting the columns of $\boldsymbol{G}'$ after calculating $\boldsymbol{G}' = \boldsymbol{S}\boldsymbol{G}_D\boldsymbol{P}$ and deleting the columns of $\boldsymbol{G}$ before calculating $\boldsymbol{G}'_D = \boldsymbol{S}\boldsymbol{G}\boldsymbol{P}$ are equivalent.

Algorithm 1: [Puncturing procedure]
Input: $k \times n$ generator matrix $\boldsymbol{G}$ of RM code
Output: $k \times (n-p)$ punctured generator matrix $\boldsymbol{G}_D$
1. Randomly pick a minimum Hamming weight code $\boldsymbol{x}$ from C.
2. Randomly pick a minimum weight codeword $\boldsymbol{y}$ from $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$.
3. Choose p, such that $\mathrm{wt}(\boldsymbol{y}) \le p \le 2\mathrm{wt}(\boldsymbol{y})$.
4. Randomly choose the set of indices L_D such that $\mathrm{supp}(\boldsymbol{y}) \subseteq L_D$ and $|L_D| = p$.
5. Delete the columns with indices in L_D from $\boldsymbol{G}$, which are denoted by $\boldsymbol{G}_D$.

3.2 Modification with puncturing and insertion

Although the modification of the RM code-based McEliece cryptosystem by insertion was proposed [7], this cryptosystem was still vulnerable to the square code attack [8]. However, it can be shown that the proposed cryptosystem, which simultaneously uses sophisticated puncturing and random insertion, can prevent the square code attack along with other known attacks.

(1) Key generation

(1-1) *Puncturing:* The random puncturing procedure is the same as that in the previous subsection by L_D, whose code is called a punctured code.

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

(1-2) *Insertion:* Let $L_{\mathrm{I}} = \{l_1, \ldots, l_I\}$ be a set of randomly chosen $|L_{\mathrm{I}}|$ column indices, where $1 \le l_i \le n - |L_D| + |L_{\mathrm{I}}|, 1 \le i \le |L_{\mathrm{I}}|$. Then, insert random columns into $\boldsymbol{G}_D$, denoted by G_{DI}, where columns with indices in L_{I} are inserted columns.

(1-3) *Generating* $\boldsymbol{S}$ *and* $\boldsymbol{P}$*:* Let $\boldsymbol{S}$ be a $k \times k$ scrambling matrix and $\boldsymbol{P}$ be an $(n - |L_D| + |L_{\mathrm{I}}|) \times (n - |L_D| + |L_{\mathrm{I}}|)$ permutation matrix. The public key is generated by calculating $\boldsymbol{G}'_{DI} = \boldsymbol{S}\boldsymbol{G}_{DI}\boldsymbol{P}$. The number of random errors, $t' = \lfloor t - |L_D|/2 \rfloor$, of $\boldsymbol{G}'_{DI}$ is known to others together with G'_{DI} being the public keys. The private keys are $\boldsymbol{P}$, $\boldsymbol{S}$, $\boldsymbol{G}$, L_D, and L_{I}.

(2) Encryption

Alice encrypts a message $m \in \{0,1\}^k$ using Bob's public key (G'_{DI}, t'). She chooses a random error vector $\boldsymbol{e} \in \{0,1\}^{n-|L_D|+|L_{\mathrm{I}}|}$ with a maximum Hamming weight of t' and sends the ciphertext $\boldsymbol{c} = mG'_{DI} + \boldsymbol{e}$ to Bob.

(3) Decryption

When Bob receives the ciphertext $\boldsymbol{c}$, he first multiplies $\boldsymbol{P}^{-1}$ to the right hand side of the ciphertext as $\boldsymbol{c}\boldsymbol{P}^{-1} = mSG_{DI} + \boldsymbol{e}\boldsymbol{P}^{-1}$. He deletes the inserted elements with indices in L_{I} of $\boldsymbol{c}\boldsymbol{P}^{-1}$ and then he inserts the erasure mark '?' in the jth positions, $j \in L_D$ for an erasure decoding, or randomly inserts '0' or "1" for normal error decoding of RM codes. After the decoding procedure, multiplying $\boldsymbol{S}^{-1}$ to $\boldsymbol{mS}$ensures he can recover the original message $\boldsymbol{m}$.

4 Security of the proposed cryptosystems

In this section, we discuss the efficacy of the proposed cryptosystem against the known attacks such as the Minder–Shokrollahi attack, Chizhov–Borodin attack, square code attack, and information set decoding.

4.1 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 $\boldsymbol{P}$, and $\boldsymbol{x}$ be the minimum weight code word in C. We would like to find the minimum $|L_D|$, where the Minder–Shokrollahi attack becomes ineffective. Again, note that all punctured positions are included in $\mathrm{supp}(\boldsymbol{x})$ 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 attack is to find the minimum weight code words. Let $\boldsymbol{x}'$ be a minimum weight code word of C', which is a punctured code word of $\boldsymbol{x} \in C$. We can then find $C'_{\mathrm{supp}(\boldsymbol{x}')}$. The support set of the punctured code word is denoted as

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

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

Clearly, the code lengths of $C_{\mathrm{supp}(\boldsymbol{x})}$ and $C'_{\mathrm{supp}(\boldsymbol{x}')}$ are the same and $C_{\mathrm{supp}(\boldsymbol{x})} \subseteq C'_{\mathrm{supp}(\boldsymbol{x}')}$, as all deleted positions are included in $\mathrm{supp}(\boldsymbol{x})$. Now, we are interested in the case of $C_{\mathrm{supp}(\boldsymbol{x})} \ne C'_{\mathrm{supp}(\boldsymbol{x}')}$, that is, $C_{\mathrm{supp}(\boldsymbol{x})} \subsetneq C'_{\mathrm{supp}(\boldsymbol{x}')}$, for which the Minder–Shokrollahi attack is ineffective. In the following theorem, we can determine the minimum number of punctured positions required to prevent the Minder–Shokrollahi attack.

Theorem 1: For the RM code $\mathrm{RM}(r,m)$, a minimum $|L_D| = 2^{m-2r}$ is required for $C_{\mathrm{supp}(\boldsymbol{x})} \subsetneq C'_{\mathrm{supp}(\boldsymbol{x}')}$. The support set of the minimum weight code word in $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$ is the essential punctured locations, where $\boldsymbol{x}$ is the minimum weight code word of C.

Proof: It is easy to check that $\mathrm{RM}(r, m-r) = \mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$. Thus, the minimum weight code word of $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$ is 2^{m-2r}. Let $\boldsymbol{y}$ be the minimum weight code word in $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$. Then, there exists $\boldsymbol{z} \in C$ such that

$$y = \mathrm{proj}_{\mathrm{supp}(x)}(\boldsymbol{z}). \tag{3}$$

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

Example 2: Consider an RM code $\mathrm{RM}(2,5)$. Suppose that the permutation matrix and scrambling matrix are identity matrices for simplicity. Clearly, one of the minimum weight code words is $\boldsymbol{x} = (1111111100\ldots00) \in \mathrm{RM}(2,5)$ or $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C) = \mathrm{RM}(2,3)$. Then, one of the minimum weight code words in $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$ is $\boldsymbol{y} = (10001000)$. Also, we set $L_D = \{1,5\}$ and the punctured code word as $\boldsymbol{x}$ is $\boldsymbol{x}' = (?111?11100\cdots00) = (11111100\ldots00)$., and z in (3) is

$$z = (10001000|10001000|10001000|10001000). \tag{4}$$

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

Example 3: Consider the RM code $\mathrm{RM}(1,4)$. Suppose that the permutation matrix and scrambling matrix are identity matrices for simplicity. Therefore, one of the minimum weight code words is $\boldsymbol{x} = (1111111100000000)$. Then, $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C) = \mathrm{RM}(1,3)$ and the minimum weight of $\mathrm{proj}_{\mathrm{supp}(\boldsymbol{x})}(C)$ is 4. Thus, to neutralise the Minder–Shokrollahi attack at least four components should be punctured. If we puncture fewer than four components, the attack cannot be neutralised. 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 $\boldsymbol{x}' = (111110\cdots0)$. Then, the generator matrix of $C'_{\mathrm{supp}(\boldsymbol{x}')}$ is

$$C'_{\mathrm{supp}(\boldsymbol{x}')} = (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1). \tag{5}$$

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

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{6}$$

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

Similarly, it is not difficult to check that

$$C'_{\mathrm{supp}(\boldsymbol{x}')} \subseteq \overbrace{C'_i \times C'_i \times \cdots \times C'_i}^{2^r - 1} = (C'_i)^{2^r - 1} \tag{7}$$

where (see (8)) It is difficult to correctly decompose component codes C'_i from $C'_{\mathrm{supp}(\boldsymbol{x}')}$, as many random component codes of $C'_{\mathrm{supp}(\boldsymbol{x}')}$ can form $\mathrm{RM}(r-1, m-r) + \{\boldsymbol{0}, \text{an element in the basis of } \mathrm{RM}(r, m-r)\}$.

4.2 Chizhov–Borodin's attack

The Chizhov–Borodin attack uses the property that the dual code of RM is also the RM code. For punctured RM codes, their dual codes are shortened RM codes [12]. It is not possible to recover the

$$C'_i = \mathrm{RM}(r-1, m-r) + \{\boldsymbol{0}, \text{an element in the basis of } \mathrm{RM}(r, m-r)\}. \tag{8}$$

RM codes from the shortened RM codes, as some rows and columns are deleted from the generator matrix. Therefore, the Chizhov–Borodin attack cannot be applied to the McEliece cryptosystem based on the proposed punctured RM codes.

Example 4: Let $\boldsymbol{G}$ be a generator matrix of RM(1, 3). $\boldsymbol{G}_D$ 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. We then have

$$\boldsymbol{G}_D = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \tag{9}$$

and the generator matrix of its dual code is given as

$$\boldsymbol{G}_D{}^{\perp} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \tag{10}$$

The dual code of RM(1, 3) is also RM(1, 3); however, the dual code of the punctured RM code is also the shortened code of the RM code. This means that $\boldsymbol{G}_D^{\perp}$ is the generator matrix of the shortened RM code of RM(1, 3); however, the rows deleted by shortening cannot be recovered.

4.3 Square code attack

For $m = 2r$, let us consider the RM code RM(r,m), with insertion of $|L_1|$ random columns, where the code length is $2^m + |L_1|$. Its square code becomes $\mathscr{C}^2 = \mathrm{RM}(m, m)$ with $|L_1|$ random column insertions. The minimum weight code word of RM(m,m) is 1, and there are m minimum weight code words in RM(m,m). Let $\boldsymbol{x}$ be one of the minimum weight code words in RM(m,m). Assume that $\boldsymbol{x}$ has 1 in jth position. If we delete the jth column of $\boldsymbol{G}$, then $\dim \mathscr{C}_j^2 = \dim \mathscr{C}^2 - 1$ with high probability. Thus, there are $|L_1| + m$ indices of reducing the dimensions of C by 1. Therefore, the probability of finding randomly inserted columns is $1/\binom{|L_1|+m}{|L_1|}$.

If $m = 10$ and $|L_1| = 10$, $1/\binom{|L_1|+m}{|L_1|} = 1/\binom{20}{10} \simeq 2^{-17.5}$. In addition, $O(n^5) \simeq 2^{50}$ operations are required for each case [8]. Thus, inserting ten random columns requires $2^{67.5}$ more operations to successfully neutralise a square code attack. Since the attacker must apply the Minder–Shokrollahi attack after a successful the square code attack, an attacker requires more time to complete this attack. The required complexity for operations of the square code attack on RM$(r, 2r)$ for $r = 5, 6$ is given in Table 1. Therefore, it is difficult to apply the square code attack on RM$(r, 2r)$ with the insertion of random columns.

Table 1 Required complexity for operations of the square code attack on RM$(r, 2r)$ code

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

Table 2 Comparison of the proposed cryptosystems with original cryptosystems in terms of information set decoding

	(n,k,t)	Number of punctured bits	T_j
RM(3, 10) without puncturing	(1024,176,200)	0	2^{61}
with puncturing	(1008,176,192)	16	2^{60}
RM(3, 11) without puncturing	(2048,232,420)	0	2^{82}
with puncturing	(2016,232,404)	32	2^{80}

For $m > 2r$, using the puncturing of the RM codes, we can also prevent the square code attack because (2) will no longer hold. In the case of $m = 2r$, we find that if there is a code word with weight 1 in C^2, the dimensions of C^2 can be reduced by deleting a column. Although the square code of RM$(2r, m)$ does not contain code words with a weight of 1, we can control the weight using the proposed puncturing method. It is known that the square code is RM$(2r, m)$ and the minimum weight of the square code is 2^{m-2r}. Since $2^{m-2r} \geq 2$, deleting one column does not reduce the dimensions [8]. However, puncturing more than $2^{m-2r} - 1$ columns (2) does not hold, similar to the case of RM$(r, 2r)$. Thus, the square code attack cannot be applied to the proposed McEliece cryptosystems. The minimum weight of punctured code in $\mathrm{RM}(r, m)^2$ is reduced when we puncture the code words in the support set of the minimum weight code word in RM(r,m). Additionally, the square code attack is effectively prevented when $m - 2r = 1, 2$, because the number of required puncturing bits is small.

4.4 Information set decoding attack

The information set decoding attack is based on finding k error-free bits $\boldsymbol{c}_k$ of ciphertext randomly. An attacker can choose k columns of $\boldsymbol{G}'_{\boldsymbol{DI}}$ with error free indices from the ciphertext, which is denoted by $\boldsymbol{G}'^{(\boldsymbol{k})}_{\boldsymbol{DI}}$. Then, $\boldsymbol{c}_{\boldsymbol{k}} = \boldsymbol{m} \cdot \boldsymbol{G}'^{(\boldsymbol{k})}_{\boldsymbol{DI}'} + \boldsymbol{e}_{\boldsymbol{k}}$ with $\boldsymbol{e}_{\boldsymbol{k}} = \boldsymbol{0}$, and the decryption is performed using $\boldsymbol{m} = \boldsymbol{c}_{\boldsymbol{k}} \cdot \boldsymbol{G}'^{(k)-1}_{DI}$. Lee and Brickell [13] generalised the information set decoding attack for $\boldsymbol{e}_{\boldsymbol{k}} \neq \boldsymbol{0}$, where the weight of $\boldsymbol{e}_{\boldsymbol{k}}$ can be less than or equal to a given integer j. The complexity of the attack is given as

$$W_j = T_j(k^3 + N_j k) \tag{11}$$

where $T_j^{-1} = \sum_{i=0}^{j} \binom{t}{i}\binom{n-t}{k} / \binom{n}{k}$ and $N_j = \sum_{i=0}^{j} \binom{k}{i}$. For an RM code RM($r,m$), the dimensions are $k = \sum_{i=0}^{r} \binom{m}{i}$, and let t be the bit error correctability. The minimum number of the punctured bits for the proposed cryptosystem is equal to $p = 2^{m-2r}$. Then, the number of correctable bit errors after puncturing is given as $t' = t - 2^{m-2r-1}$ and the number of columns of the generator matrix reduces to $n' = 2^m - 2^{m-2r}$. As the term $k^3 + N_j k$ is independent of n and t, it is reasonable to compare the term T_j for complexity of an information set decoding attack.

For example, consider the case of RM(3, 10). Using the decoding algorithm in [10], the number of correctable bit errors is $t = 200$. In the case of $(n, k, t) = (1024, 176, 200)$ without puncturing, the approximate value of T_j is 2^{61}. In the case of $(n', k, t') = (1008, 176, 192)$ with puncturing, we have $T_j \simeq 2^{60}$. In Table 2, for the cases of RM(3, 10) and RM(3, 11) with or without puncturing, the corresponding values of T_j are compared.

Table 1 provides the computational complexity of the information set decoding is slightly reduced after column puncturing. Thus, the effect of puncturing columns of the generator matrix for the proposed cryptosystems should be minimised.

5 Conclusions

In this paper, we proposed secure modification methods for the McEliece cryptosystem based on the punctured RM codes. We found the exact number and locations of puncturing of the generator matrix of the original RM codes to prevent the aforementioned attacks. While the previous McEliece cryptosystem based on RM codes is vulnerable to known attacks such as the Minder–Shokrollahi attack, the Chizhov–Borodin attack, and square code attack, the proposed punctured RM code-based McEliece cryptosystem can repel these attacks. While security is slightly reduced as a result of puncturing, the proposed cryptosystem can be revived.

6 Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (R-20160229-002941, Research on Lightweight Post-Quantum Crypto-systems for IoT and Cloud Computing).

7 References

[1] McEliece, R.J.: 'A public-key cryptosystem based on algebraic coding theory'. DSN Progress Report, DSN PR 42-44, 1978, pp. 114–116

[2] Sidelnikov, V.M., Shestakov, S.O.: 'On insecurity of cryptosystems based on generalized Reed–Solomon codes', *Discrete Math. Appl.*, 1992, **1**, (4), pp. 439–444

[3] Shrestha, S.R., Kim, Y.-S.: 'New McEliece cryptosystem based on polar codes as a candidate for post-quantum cryptography'. Proc. ISCIT 2014, Incheon, Korea, 24–26 September 2014, pp. 368–372

[4] Sidelnikov, V.M.: 'A public-key cryptosystem based on binary Reed–Muller codes', *Discrete Math. Appl.*, 1994, **4**, (3), pp. 191–207

[5] Minder, L., Shokrollahi, A.: 'Cryptanalysis of the Sidelnikov cryptosystem'. Proc. EUROCRYPT 2007, 2007 (LNCS, **4515**), pp. 347–360

[6] Couvreur, A., Gaborit, P., Gauthier-Umana, V.*, et al.*: 'Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes', *Des. Codes Cryptogr.*, 2014, **73**, (2), pp. 641–666

[7] Gueye, C.T., Mboup, E.H.M.: 'Secure cryptographic scheme based on modified Reed–Muller codes', *Int. J. Secur. Appl.*, 2013, **7**, (3), pp. 55–64

[8] Otmani, A., Kalachi, H.T.: 'Square code attack on a modified Sidelnikov cryptosystem'. Proc. C2SI, 2015 (LNCS, **9084**), pp. 173–183

[9] Esmaeili, M., Dakhilalian, M., Gulliver, T.A.: 'New secure channel coding scheme based on randomly punctured quasi-cyclic low-density parity check codes', *IET Commun.*, 2014, **8**, (14), pp. 2556–2562

[10] Sidelnikov, V.M., Pershakov, A.S.: 'Decoding of Reed–Muller codes with a large number of errors', *Probl. Inf. Transm.*, 1993, **28**, (3), pp. 269–282 (A Translation of Problemy Peredachi Informatsii)

[11] Chizhov, I.V., Borodin, M.A.: 'The failure of McEliece PKC based on Reed–Muller codes', *Prikl. Diskr. Mat. Suppl.*, 2013, **6**, pp. 48–49

[12] Boyle, E.C., McEliece, R.J.: 'Asymptotic weight enumerators of randomly punctured, expurgated, and shortened code ensembles'. Proc. 46th Annual Allerton Conf. Communication, Control, and Computing, September 2008, pp. 910–917

[13] Lee, P., Brickell, E.: 'An observation on the security of McEliece's public key cryptosystem'. Proc. EUROCRYPT'88, 1989 (LNCS, **330**), pp. 275–280