Hindawi
Security and Communication Networks
Volume 2019, Article ID 3086975, 10 pages
https://doi.org/10.1155/2019/3086975

Research Article

polarRLCE: A New Code-Based Cryptosystem Using Polar Codes

Jingang Liu,[1] Yongge Wang,[2] Zongxiang Yi,[1] and Zhiqiang Lin[1]

[1]*School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China*
[2]*UNC Charlotte, Charlotte, NC 28223, USA*

Correspondence should be addressed to Zhiqiang Lin; linzhiqiang0824@163.com

Received 20 September 2019; Accepted 27 November 2019; Published 26 December 2019

Guest Editor: Andrea Visconti

Security challenges brought about by the upcoming 5G era should be taken seriously. Code-based cryptography leverages difficult problems in coding theory and is one of the main techniques enabling cryptographic primitives in the postquantum scenario. In this work, we propose the first efficient secure scheme based on polar codes (i.e., *polarRLCE*) which is inspired by the RLCE scheme, a candidate for the NIST postquantum cryptography standardization in the first round. In addition to avoiding some weaknesses of the RLCE scheme, we show that, with the proper choice of parameters, using polar codes, it is possible to design an encryption scheme to achieve the intended security level while retaining a reasonably small public key size. In addition, we also present a KEM version of the polarRLCE scheme that can attain a negligible decryption failure rate within the corresponding security parameters. It is shown that our proposal enjoys an apparent advantage to decrease the public key size, especially on the high-security level.

1. Introduction

Cryptography is essential for the security of online communication. However, many commonly used cryptosystems will be completely broken once large quantum computers exist. It is well known that several computation-intensive tasks may be significantly accelerated through algorithms running on a quantum computer, such as Shor's [1] and Grover's [2] algorithm. Current cryptographic protocols, such as RSA and Diffie–Hellman, are proven to be vulnerable under quantum algorithms. This fact pushed cryptographic research to focus on postquantum solutions, i.e., finding new primitives based on more well-suited mathematical problems that may still be difficult to solve for a quantum computer. With this in mind, the US National Institute of Standards and Technology (NIST) is now beginning to prepare for the transition into postquantum cryptography (PQC) and has launched a call for PQC standardization project [3], and this ongoing standardization has moved on to 2nd round thus far. Due to its inherent resistance to attacks by quantum computers, code-based cryptography is one of the main candidates for the PQC standardization call, alongside multivariate and lattice-based schemes.

Code-based cryptography is accepted as quantum computing resistant based on a hard coding theory problem, decoding a random linear code in some metric. Historically, the conservative and well-understood choices for code-based cryptography are the McEliece cryptosystem [4] and its dual variant by Niederreiter [5] using binary Goppa codes. However, they suffer from the disadvantage of having large public key size, in spite of the fast encryption and decryption operations. It is therefore of utmost importance to seek ways to reduce the key sizes for code-based cryptosystems while keeping their security level. After the original proposal of the code-based encryption scheme by McEliece [4] which was based on binary Goppa codes, several variants have been proposed using different codes that allow for smaller keys or more efficient encoding and decoding algorithms, e.g., algebraic geometric (AG) codes [6], generalized Reed–Solomon (GRS) codes [7, 8], low-density parity check (LDPC) codes [9, 10], Reed–Muller (RM) codes [11], low-rank parity check (LRPC) codes [12], and among others. Although the original McEliece cryptosystem remains secure, most of these variants have been successfully cryptanalyzed [13–17]. Despite their promising features, the alternative codes need to be handled carefully due to too much structure.

It is noteworthy that Wang [18, 19] proposed a random linear code-based quantum resistant public key encryption scheme, referred as RLCE, which is a variant of the McEliece encryption scheme. They analyzed an instantiation of the RLCE scheme using GRS codes and introduced randomness in public key, which is based on the juxtaposition of a GRS code with a random linear code. The idea of the RLCE scheme is to use a distortion matrix that mixes some random columns with the structured ones. The advantage of the RLCE scheme is that its security does not depend on any specific structure of underlying linear codes, instead, it is based on the $\mathcal{NP}$-hardness of decoding random linear codes. In such a manner, previous attacks regarding GRS codes based on the technique of filtration distinguisher no longer work. Nevertheless, part of the original parameters was attacked by [20], and RLCE was not selected for the second round of the NIST PQC standardization call.

Polar codes, introduced by Arikan in [21], have received much attention since they are the first class of error-correcting codes that provably achieve the capacity for any symmetric binary discrete memoryless channel (B-DMC) with very efficient encoding and decoding algorithm, whose time complexity scales as $\mathcal{O}(n \log n)$, where n is the length of the code. Because of the good performance and low complexity, polar codes have been adopted for use in future wireless communication systems (e.g., 5G cellular systems). Looking forward, there is a critical need to ensure that 5G techniques, as developed, envision future adoption of PQC for public key cryptosystems.

1.1. Related Work. Within this thread of research, there are two heuristic variants [22, 23] of the McEliece cryptosystem based on polar codes. The first one [23] was broken by Bardet et al. [24] using the structure of the minimum weight codewords. They managed to solve the code equivalence problem for polar codes and thus completely broke the scheme [23] based on polar codes. The second variant was presented by Hooshmand et al. [22] which suggested using the subcode of polar codes. However, we found that the proposal in [22] is useless in practice since 80% ciphertexts could not be decrypted, as discussed in Section 2.4.

1.2. Our Contribution. In this work, we combine the idea of the RLCE scheme by inserting random columns, then propose the first efficient secure scheme based on polar codes (i.e., *polarRLCE*), which can avoid the attack of [24]. Furthermore, possible attacks are outlined and the key size of several choices of parameters is compared to those of known schemes with the same security level. We show that the existing attacks on the proposal scheme do not seem to be effective. More importantly, our proposal enjoys an apparent advantage to decrease the public key size, especially on the high-security level. It allows us to reconsider polar codes as a good candidate for using in code-based cryptography.

The rest of this paper is organized as follows. Some necessary preliminaries such as notation and definitions are given in the Section 2. In Section 3, we present the precise description of the construction of the polarRLCE scheme. Section 4 discusses the known cryptanalytic attacks against our proposal and presents a compact key-encapsulation mechanism (KEM) version regarding polarRLCE. Furthermore, we give the choice of suggested parameters and key size for the achievable security level. Finally, some concluding remarks are made in Section 5.

2. Preliminaries

In this section, we introduce some of the basic background information necessary to follow this paper. Throughout the paper, we will denote vectors by lower-case bold letters, e.g., $\mathbf{m}$. And denote matrices by upper-case bold letters, e.g., $\mathbf{A}$.

2.1. Coding Theory. We begin by briefly reviewing the basic concepts in coding theory and show its application to public-key cryptography.

Definition 1 (linear codes). An $[n, k]$ linear code $\mathscr{C}$ over a finite field $\mathbb{F}_q$ is a k-dimensional linear subspace of $\mathbb{F}_q^n$.

Definition 2 (generator matrix and parity check matrix). A $k \times n$ matrix $\mathbf{G}$ with entries from $\mathbb{F}_q$ having row-span $\mathscr{C}$ is a generator matrix for the $[n, k]$ linear code $\mathscr{C}$. And parity check matrix $\mathbf{H}$ is a $(n-k) \times n$ matrix whose rows generate the orthogonal complement of $\mathscr{C}$.

One can specify a linear code $\mathscr{C}$ via a generator matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ or a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ via

$$\mathscr{C} := \left\{ \mathbf{xG} \in \mathbb{F}_q^n \,\middle|\, \mathbf{x} \in \mathbb{F}_q^k \right\} \text{ or } \mathscr{C} := \left\{ \mathbf{c} \in \mathbb{F}_q^n \,\middle|\, \mathbf{Hc}^T = 0 \right\}. \quad (1)$$

If $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ or $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, i.e., each matrix entry is chosen uniformly at random from $\mathbb{F}_q$, then we call $\mathscr{C}$ a random linear code.

The code $\mathscr{C}$ can be represented by different generator matrices. An important one is the systematic form, i.e., when each input symbol is directly represented in its first k coordinate positions. For a systematic linear code, the generator matrix $\mathbf{G}$ can always be written as $\mathbf{G} = (\mathbf{I}_k \mathbf{P})$, where $\mathbf{I}_k$ is the identity matrix of size k. And if $\mathbf{G}$ has such a systematic form, then $\mathbf{H} = (-\mathbf{P}^T \mathbf{I}_{n-k})$.

Definition 3 (punctured and shortened codes). Given an $[n, k]$ linear code $\mathscr{C}$, let I be a subset of $\{1, \ldots, n\}$ and the ith entry of a codeword $\mathbf{c} \in \mathscr{C}$ is written as $\mathbf{c}_i$. Then, we define the punctured code $\mathscr{P}_I(\mathscr{C})$ and the shortened code $\mathscr{S}_I(\mathscr{C})$ as

$$\begin{aligned} \mathscr{P}_I(\mathscr{C}) &= \left\{ (\mathbf{c}_i)_{i\notin I} \,\middle|\, \mathbf{c} \in \mathscr{C} \right\}, \\ \mathscr{S}_I(\mathscr{C}) &= \left\{ (\mathbf{c}_i)_{i\notin I} \,\middle|\, \exists \mathbf{c} \in \mathscr{C}, \text{s.t. } \forall i \in I, \mathbf{c}_i = 0 \right\}. \end{aligned} \quad (2)$$

Given a subset I of the set of coordinates of a vector $\mathbf{x}$, we denote by $\mathscr{P}_I(\mathbf{x})$ the vector $\mathbf{x}$ punctured at I, that is to say, whose ith entry has been deleted for any $i \in I$.

Lemma 1. *Let $\mathscr{C}$ be a code of dimension k and generator matrix $\mathbf{G}$. Then, the matrix $\mathbf{G}_{\mathscr{P}}$ is a generator matrix for $\mathscr{P}_I(\mathscr{C})$, which can be obtained by deleting the columns from $\mathbf{G}$ index in I.*

We now only consider binary codes, i.e., $q = 2$. The hamming weight $w_H(\mathbf{x})$ of a binary vector in $\mathbf{x} \in \mathbb{F}_2^n$ is the number of nonzero entries in the vector. And the minimum hamming distance of the code $\mathscr{C}$ is defined as $d = \min\{w_H(\mathbf{x} - \mathbf{y})\}$, where $\mathbf{x} \neq \mathbf{y}$.

2.2. McEliece's Public-Key Cryptosystem. In 1978, McEliece presented in his seminal paper [4] the first code-based public key encryption system, which relied on Goppa codes to form the secret key. And a permutation matrix and an invertible matrix are used for scrambling and concealing secret key. It employs an $[n, k]$ linear code $\mathscr{C}$ over $\mathbb{F}_2$, with an error-correcting capability of t errors, for which an efficient decoding algorithm is known. The general key generation, encryption, and decryption steps for the original proposal in [4] work as follows.

Private key: it consists of three matrices $\mathbf{G}$, $\mathbf{S}$, and $\mathbf{P}$, where $\mathbf{G}$ is an $k \times n$ generator matrix of this code, $\mathbf{S}$ is an arbitrary $k \times k$ binary nonsingular matrix (called the scrambling matrix), and $\mathbf{P}$ is an $n \times n$ random permutation matrix.

Public key: it is composed of the $k \times n$ matrix $\mathbf{G}'$ defined by $\mathbf{G}' = \mathbf{SGP}$ and the error-correcting capability with t.

Encryption: to encrypt the message $\mathbf{m} \in \mathbb{F}_2^k$ and choose a random error vector $\mathbf{e} \in \mathbb{F}_2^n$ with weight $w_H(\mathbf{e}) \leq t$, then the corresponding ciphertext is computed as

$$y = \mathbf{m}\mathbf{G}' + \mathbf{e}. \tag{3}$$

Decryption: the decryption procedure consists in computing

$$y\mathbf{P}^{-1} = \mathbf{mSG} + \mathbf{e}\mathbf{P}^{-1}, \tag{4}$$

and using a fast decoding algorithm for Goppa code $\mathscr{C}$ to recover $\mathbf{mS}$. The message is then recovered by $\mathbf{m} = (\mathbf{mS})\mathbf{S}^{-1}$.

Notice that multiplying the error vector by a permutation does not change the weight of the vector. One can easily verify the correctness of the scheme by checking

$$\text{Decrypt}(\text{Encrypt}(\mathbf{m}, \text{pk}), \text{sk}) = \mathbf{m}. \tag{5}$$

A dual version of the McEliece cryptosystem that uses the parity-check matrix instead of the generating matrix has been proposed by Niederreiter in [5]. Following the idea of [25], the Niederreiter system and the McEliece system are equivalent in terms of security.

Knowing the description of the selected Goppa code $\mathscr{C}$ allows efficient decoding, since there are many efficient decoding algorithms for this problem running in polynomial time. However, knowing only the public key $\mathbf{G}'$, the attacker is facing a decoding problem for a code that looks like a random code, which is $\mathscr{NP}$-hardness. The attacker can either try to decode an intercepted ciphertext (message recovery attack) or try to recover the secret matrix $\mathbf{G}$ from the public matrix $\mathbf{G}'$ (key-recovery attack).

The security level of the McEliece system has remained remarkably stable, despite dozens of attack papers over 40 years, regardless of the original McEliece parameters being designed for only 64-bit security level. For instance, as recommended by Bernstein et al. [26], the McEliece scheme with binary Goppa codes using code length $n = 2960$ and code dimension $k = 2288$ and adding $t = 57$ errors can achieve 128-bit security level, Thus, the corresponding public key size is 187.69 KBytes.

2.3. Polar Codes Construction. We first recall the basic facts about polar codes. As shown in the seminal work by Arikan [21], for any B-DMC, there exists a polar code of block length $n = 2^m$ which is characterized by the information bit set $\mathscr{A}$ with exponentially small word-error rate under successive cancellation (SC) decoder. A polar code may be specified completely by $(n, k, \mathscr{F})$, where n is the length of a codeword in bits, k is the number of information bits encoded per codeword, and $\mathscr{F}$ is a set of $n - k$ integer indices called frozen bit locations from $\{0, 1, \ldots, n-1\}$. The k more reliable subchannels (based on the polarization phenomenon) with indices in set $\mathscr{A}$ carry information bits and the rest subchannels included in the complementary set $\mathscr{A}^c$ (i.e., the set $\mathscr{F}$) can be set to fixed bit values, such as all zeros. Generally, the challenge is to select the information bits set $\mathscr{A}$ or, more precisely, the methods that are proposed for finding the indices of good polarized channels.

For a binary polar code of length $n = 2^m$, the polar encoding of an input vector is carried out by the polarization transformation matrix $\mathbf{G}_n = \mathbf{F}^{\otimes m}$, which is the mth Kronecker power of the 2×2 kernel matrix:

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

For a given noise channel, the generator matrix $\mathbf{G}$ of an $[n, k]$ polar code is defined as the submatrix of $\mathbf{G}_n$ consisting of k rows with indices corresponding to information set $\mathscr{A} = \{i_1, i_2, \ldots, i_k\}$. Roughly speaking these rows are chosen in such a way that it gives good performance for the SC decoder. These codes come equipped with an SC decoder whose decoding complexity scales as $\mathcal{O}(n \log n)$ (see [21] for more details).

The idea of exploiting polar codes in cryptography came in a natural way since polar codes benefit of various interesting properties: can achieve Shannon capacity for the class of binary discrete memoryless channels, attain better performance (lower decoding errors) because of the channel polarization along with the increased block length, posses efficient encoding and decoding procedures, etc. Even though polar codes are closely related to RM codes, the techniques used for the cryptanalysis of RM codes do not work on polar codes.

2.4. The Proposal by Hooshmand et al. [22]. The error-correcting capacity of polar codes [21] does not only depend on the code length but also on other factors such as the code rate and the designed channel. However, the error-correcting capacity was merely set to be a fixed value of $t = 2\sqrt{n} - 1$ in the proposal by Hooshmand et al. [22], they did not consider the error probability of decoding. For instance, they claimed that one can use $[2048, 1750]$-polar code with $t = 89$, which followed Theorem 8 in [27]. Actually, Theorem 8 from [27] is only suitable for the concatenated polar codes with respect to the length of burst-errors as stated in [27]. Nevertheless, the proposal in [22] used random errors through the encryption process. With these parameters in [22], we performed numerical simulation using MATLAB R2018a where 10^5 decoding trails are exploited under SC decoder [21], and the experiment result indicates that the decoding error probability is nearly 0.8, i.e., 80% ciphertexts could not be decrypted and cannot be employed in a practical environment. With respect to our proposal, the error probability is approximately 2^{-14} (see Section 3). Furthermore, we transform the basic polarRLCE into a key-encapsulation mechanism (KEM) version, which can achieve the negligible decryption failure rate (DFR) within the corresponding security parameters.

3. Our Proposed Scheme of polarRLCE

In this section, we describe our new variant of the McEliece cryptosystem by exploiting the method of RLCE [18, 19] scheme. More precisely, the procedures of our polarRLCE are specified as follows.

Key generation: according to the construction of the polar code in Section 2.3,

(i) Choose an $[n, k]$ polar code with the generator matrix $\mathbf{G}$ of length n and dimension k.

(ii) Generate w random column vectors $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_w$, and let

$$\mathbf{G}_1 = (\mathbf{g}_1, \ldots, \mathbf{g}_{n-w}, \mathbf{g}_{n-w+1}, \mathbf{r}_1, \ldots, \mathbf{g}_n, \mathbf{r}_w), \tag{7}$$

be the $k \times (n + w)$ matrix obtained by inserting w random $k \times 1$ column vectors $\mathbf{r}_i$ into matrix $\mathbf{G}$.

(iii) To mix the columns, choose w random nonsingular binary 2×2 matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_w$. Denote $\mathbf{A}$ with the $(n + w) \times (n + w)$ block-diagonal matrix:

$$\mathbf{A} = \begin{pmatrix} \mathbf{I}_{n-w} & & (0) \\ & \mathbf{A}_1 & \\ & & \ddots \\ (0) & & \mathbf{A}_w \end{pmatrix}. \tag{8}$$

(iv) Let $\mathbf{S}$ be a randomly chosen $k \times k$ nonsingular matrix and $\mathbf{P}$ be the $(n + w) \times (n + w)$ permutation matrix.

(v) The public key $k \times (n + w)$ matrix is defined as

$$\mathbf{G}_{\text{pub}} = \mathbf{S}\mathbf{G}_1\mathbf{A}\mathbf{P}. \tag{9}$$

Then, the public key and private key are given, respectively, by

$$\mathbf{G}_{\text{pub}} \text{ and } (\mathbf{G}, \mathbf{S}, \mathbf{P}, \mathbf{A}). \tag{10}$$

Encryption: let $\mathbf{m} \in \mathbb{F}_2^k$ be the message to be encrypted. Then, we randomly generate error vector $\mathbf{e} \in \mathbb{F}_2^{n+w}$ such that the hamming weight $w_{\text{H}}(\mathbf{e}) \leq t$. Compute the corresponding ciphertext:

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e}. \tag{11}$$

Decryption: to decrypt the received ciphertext $\mathbf{c}$,

(i) Calculate $\mathbf{c}\mathbf{P}^{-1}\mathbf{A}^{-1} = (c'_1, c'_2, \ldots, c'_{n+w})$.

(ii) Delete w entries at the $\mathbf{r}_i$ position of row vector $(c'_1, c'_2, \ldots, c'_{n+w})$. We denote the obtained n-length vector by

$$c' = (c'_1, c'_2, \ldots, c'_{n-w+1}, c'_{n-w+3}, c'_{n-w+5}, \ldots, c'_{n+w-1}). \tag{12}$$

(iii) It is easy to check that $c' = \mathbf{m}\mathbf{S}\mathbf{G} + e'$ for some error vector $e' \in \mathbb{F}_2^n$, where $w_{\text{H}}(e') \leq t$. Then, using the efficient decoding algorithm, one can recover the corresponding message $\mathbf{m}$.

For the purpose of constructing polar code used in our proposed variant scheme, we consider here the binary symmetric channel (BSC) with crossover probability $\varepsilon = 0.05$. For instance, to achieve 128-bit security, for reliable decoding and keeping reasonably small key size with enough security level, we will set the choice of parameters such that $n = 2^{11}$, $k = 500$, and $w = 50$. Following the method of Dragoi [28], validated through exhaustive simulation, we can choose the error vector weight of $t = 285$ with the reasonable decoding error probability is approximately 2^{-14}.

Remark 1. Please note that our scheme allows occasional decryption failures for valid ciphertexts (similar to some NIST PQC submissions), which is inherited from the decoding algorithm. However, for the good performance of polar codes, one can easily resolve this issue through repeated encryption as presented by Eaton et al. [29] which can reduce the decryption failure rate to a level negligible in the security parameter, without altering the whole parameters.

4. Security Analysis

In this section we will discuss several possible attacks against our proposed polarRLCE scheme in 3. There are two main attacks to thwart, i.e., key structural attack and decoding attack.

Furthermore, if the code $\mathscr{C}$, whose generator matrix is used as a part of the public key, could be distinguished, then an adversary could exploit the structure of $\mathscr{C}$, and this would also possibly allow the adversary to develop faster attacking algorithms. Indeed, most of these variations of the McEliece system are vulnerable to structural attacks because of the algebraic structure of underlying codes.

4.1. Brute Force Attack. A brute force attack is a trial-and-error method used to obtain the correct keys. For our proposed scheme, recall that the private key $(\mathbf{G}, \mathbf{S}, \mathbf{P}, \mathbf{A})$ is obtained randomly. Moreover, the number of candidate invertible scrambling matrix $\mathbf{S}_{k\times k}$ over $\mathbb{F}_2$ is

$$\mathscr{N} = \prod_{i=1}^{k}\left(2^k - 2^{i-1}\right) = 2^{k^2}\prod_{i=1}^{k}\left(1 - 2^i\right) > 2^{k^2-2}. \tag{13}$$

By putting in the suggested parameters (with 128-bit security) $n = 2048$, $k = 500$, and $w = 50$, it turns out that it is as well infeasible to retrieve the other three elements building the private key just by guessing, since there exist $(n + w)! = 2048! \gg 2^{128}$ different matrix $\mathbf{P}$ and nearly $\mathscr{N} = 2^{500^2} \gg 2^{128}$ choices for $\mathbf{S}$. Moreover, the candidate of block-diagonal matrix $\mathbf{A}$ is $6^w = 2^{129.25}$. Hence, the complexity of the exhaustive search attack against our scheme has an exponential time, which indicates this attack is impractical.

4.2. Square Attack. In this section, we study the square attack on our polarRLCE. There has been an increased interest in the square (a.k.a. Schur product) of linear codes in the last years (cf. [30]). Another and more recent application of the Schur product concerns cryptanalysis of code-based public key cryptosystems. In this context, the Schur product is a very powerful operation which can help to distinguish secret codes from random ones.

In fact, the method of inserting random columns or rows in the secret matrix has indeed proposed [7, 8, 31] to avoid structural attacks on similar versions of the McEliece cryptosystem. Although this proposal has effectively avoided the original attack, recent studies [14, 32, 33] have shown that in the case of GRS codes or RM codes, the random columns can be found through the consideration of the dimension of the Schur product code.

Definition 4 (Schur product). Let $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, then the Schur product of two vectors is denoted by

$$\mathbf{x} * \mathbf{y} = (x_1 y_1, x_2 y_2, \ldots, x_n y_n). \tag{14}$$

Definition 5 (square code). Let $\mathscr{A}$ and $\mathscr{B}$ be linear codes with length n. The Schur product of the two codes is the vector space spanned by all products $\mathbf{a} * \mathbf{b}$ with $\mathbf{a} \in \mathscr{A}$ and $\mathbf{b} \in \mathscr{B}$:

$$\langle \mathscr{A} * \mathscr{B} \rangle = \langle \{\mathbf{a} * \mathbf{b} \mid \mathbf{a} \in \mathscr{A} \text{ and } \mathbf{b} \in \mathscr{B}\} \rangle. \tag{15}$$

If $\mathscr{A} = \mathscr{B}$, then we call $\langle \mathscr{A} * \mathscr{A} \rangle$ the square code of $\mathscr{A}$ and denote it by $\langle \mathscr{A}^2 \rangle$. The impact of the square code on the code-based cryptosystem becomes clear when we study the dimension of these constructions.

Definition 6 (Schur matrix). Let $\mathbf{G}$ be a $k \times n$ matrix, with rows $(g_i)_{1\le i\le k}$. The Schur matrix of $\mathbf{G}$, denoted by $S(\mathbf{G})$ consists of the rows $g_i * g_j$ for $1 \le i \le j \le k$.

We observe that if $\mathbf{G}$ is a generator matrix of a code $\mathscr{C}$, then its Schur matrix $S(\mathbf{G})$ (or the submatrix which contains the linear independent rows of $S(\mathbf{G})$) is a generator matrix of the square code of $\mathscr{C}$. For the $k \times n$ matrix $\mathbf{G}$, the matrix $S(\mathbf{G})$ at most has the size $\binom{k+1}{2} \times n$ (refer to [30]).

It is well known that the square of a linear code $\mathscr{C}[n, k]$ has the dimension

$$\dim\left(\mathscr{C}^2\right) \le \min\left\{n, \frac{1}{2}k(k+1)\right\}, \tag{16}$$

and a random linear code attains this upper bound with high probability.

One of the key features in most of the successful cryptanalysis efforts has been that the proposed codes have small Schur-product dimension which leads to key recovery or distinguishing attacks. In particular, this lends credence to the idea that codes with small Schur-product dimension appear to be unsuitable for use in the McEliece framework. For instance, if the code is generalized Reed–Solomon (GRS) code, then it satisfies

$$\dim\left(\mathscr{C}^2\right) = \min\{n, 2k - 1\}, \tag{17}$$

and fulfills this lower bound with equality, i.e., for $k < (n/2)$, their square dimension is much smaller than one expects from a random code. Actually, this fact is, e.g., utilized by [14, 33] to build an effective distinguisher, yielding a structural attack on the GRS-based McEliece cryptosystem.

Looking at the definition of the square code, we observe that it is generated by all possible Schur-products of every pair of (nonnecessarily distinct) codewords in the given linear code. Therefore, it is natural to expect that the dimension of the square code is "as large as possible." In other words, for a randomly chosen linear code $\mathscr{R}$, we expect that inequality (16) is actually an equality with very high probability.

Let us consider the recent work [34] which reported that it might possibly exist as a heuristic distinguisher, if given two specific weakly decreasing sets. However, in the case of our polarRLCE scheme, such sets could not be found easily because of the extended public codes by inserting random columns.

To illustrate the square attack, we performed simulation by generating 10,000 random sets of the public key matrix. Our experimental result shows that, as in the case of the proposed polarRLCE scheme, the square code of the public code can always reach the maximal dimension bound. Considering the choice of parameters (with 128-bit security) such that $n = 2048$, $k = 500$, and $w = 50$. So, we can obtain the $k \times (n + w)$ public key matrix $\mathbf{G}_{\text{pub}}$. Denote the extended public code as $\mathscr{C}_{\text{pub}}$. Hence, from inequality (16), we have

$$\dim\left(\mathscr{C}_{\text{pub}}^{2}\right) = \dim\left(S\left(\mathbf{G}_{\text{pub}}\right)\right) \leq \min\left\{n+w, \frac{1}{2}k(k+1)\right\}. \tag{18}$$

For the proposed parameters, we observed experimentally that the dimension of the public matrix by the square product always reach maximum, that is to say,

$$\dim\left(S\left(\mathbf{G}_{\text{pub}}\right)\right) = n + w = 2098. \tag{19}$$

Furthermore, after perform random puncturing operations, $\mathscr{P}_I(\mathbf{G}_{\text{pub}})$, alternatively, we can obtain

$$\dim\left(S\left(\mathscr{P}_I\left(\mathbf{G}_{\text{pub}}\right)\right)\right) = n + w - |I|. \tag{20}$$

On the basis of the observations made as stated above, we claim that the technique of square attack regarding our polarRLCE could not be used to distinguish from random codes.

4.3. Key-Recovery Attack. The key-recovery attack is one of the important ways of structural attack, consists in recovering the private key from the public key. In this case, the methods are specific to the code family. In order to compute the private key of a given public key, it is often reduced to solve the code equivalence problem.

Definition 7. Let $\mathbf{G}$ and $\mathbf{G}^*$ be the generating matrix for two $[n, k]$ binary linear codes. Given $\mathbf{G}$ and $\mathbf{G}^*$, there exist a $k \times k$ binary invertible matrix $\mathbf{S}$ and $n \times n$ permutation matrix $\mathbf{P}$ such that $\mathbf{G}^* = \mathbf{SGP}$?.

This problem was first studied by Petrank and Roth [35] over the binary field. And the most common algorithm used to solve this problem is the support splitting algorithm (SSA) [36]. SSA is very efficient in the random case, but it cannot be used in the case of codes with large hulls or codes with large permutation group such as Goppa codes and polar codes.

However, a very effective structural attack on the variant [23] using polar codes was introduced by Bardet et al. in [24]. Firstly, they managed to determine exactly the structure of the minimum weight codeword of the original polar codes. Then, they solved the code equivalence problem for polar codes with respect to decreasing monomial codes. Notice that this attack is very specific to the simple usage of polar codes in [23].

Regarding our proposal, there is a really effective way of protecting the scheme since the structure of the private code is someway shattered by inserting random elements. Thus, even though one can find enough low-weight codewords, while they are not subject to the original polar code, because that these codewords possess an extended length $n + w$ which is generated by the public key matrix $\mathbf{G}_{\text{pub}}$. The natural way is to perform puncturing operations, but an exponential number of codewords need to check since there are

$$\binom{n+w}{w} = \binom{2098}{50} = 2^{336.68}. \tag{21}$$

On the other hand, the adversary cannot identify the inserted positions by distinguishing attack or square attack as stated on Section 4.2. So, the code equivalence problem becomes even more complicated to solve in this case. Therefore, the attack by [24, 34] does not apply directly to our proposed polarRLCE scheme.

Finally, we notice that very recently, Couvreur et al. [20] presented a key-recovery attack on half the parameter sets proposed in the RLCE scheme [19]. They showed that it is possible to distinguish some keys from random codes by computing the square of some shortened public codes. The set of positions $\{1, \ldots, n+w\}$ splitted into four parts based on the fact that the entry of any GRS codeword satisfies a specific expression formalization, i.e., $\dim \mathrm{GRS}_k(x, y)^2 = 2k - 1$, and then recognizes the twin positions. While polar codes are used for the aforementioned situation because of the different structure between polar codes and GRS codes.

According to the aforementioned analysis and the fact that we found no other distinguishing methods for our proposal, we claim that it is indeed able to avoid the key-recovery attack.

4.4. Message-Decoding Attack. Message-decoding attack is an important issue in code-based cryptography. The problem of recovering the private message from a ciphertext is directly related to the hardness of generic decoding for the linear code. One possibility attack to recover the message is information set decoding (ISD) algorithm, which means to decode a random linear code without exploiting any structural property of the code. The ISD algorithm searches for an information set such that the error positions are all out of the information set. The work factor of ISD clearly increases with the number of errors added in the encryption process. Thus, when choosing parameters, we will focus mainly on defeating attacks of the ISD family.

This technique was first introduced by Prange [37]. Hereafter, numerous different algorithmic techniques have been explored to improve complexity of ISD algorithm. Among several variants [26, 38–40] and generalizations, it is noteworthy that most modern ISD algorithms are based on Stern's [38] algorithm, which incorporates collision search methods to speed up decoding.

Thus, we will move on to analyze the complexity of Stern's algorithm. Similarly, they try to find a t-weight codeword in an $[n, k]$ linear code $\mathscr{C}$ generated by $\mathbb{F}_2^{k\times n}$. More precisely, apart from the generator matrix $\mathbf{G}$, the algorithm takes as input additional integer parameters p and l such that $0 \leq p \leq t$ and $0 \leq l \leq n - k$. Each iteration consists of the steps described in Algorithm 1.

$$\mathscr{P}_{\text{stern}} = \frac{\binom{k/2}{p}^2 \binom{n-k-l}{t-2p}}{\binom{n}{t}}. \tag{22}$$

Then, the probability of success in one single iteration is

And the cost of one iteration of Stern's algorithm is as follows:

Input: Generator matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$, with parameters t, p, and l.
Output: Codeword $\mathbf{c} \in \mathscr{C}$, s.t. $w_H(\mathbf{c}) = t$.
1 Select a random information set $\mathscr{I}$ from $\{1, 2, \ldots, n\}$ and divide it into two equal size subsets $\mathscr{X}$ and $\mathscr{Y}$. Moreover, select a size-l subset $\mathscr{Z}$ of $\{1, 2, \ldots, n\}\backslash\mathscr{I}$.
2 Permutate $\mathbf{G}$ randomly, and let $\mathbf{G}_{\text{sys}}$ denote the systematic form: $\mathbf{G}_{\text{sys}} = (\mathbf{IQJ})$
where $\mathbf{I}$ is the $k \times k$ identity matrix, $\mathbf{Q}$ is a $k \times l$ matrix and $\mathbf{J}$ is a $k \times (n-k-l)$ matrix.
3 Let $\mathbf{u}$ run through all p-weight vectors of length $k/2$. Then, put all vectors $\mathbf{x} = (\mathbf{u}0)\mathbf{G}_{\text{sys}}$ in a sorted list $\mathscr{L}_1$, sorted according to index $\phi(\mathbf{x})$, with $\phi(\mathbf{x})$ being the value of a vector $\mathbf{x}$ in positions $k+1$ to $k+l$, i.e., $\phi(\mathbf{x}) = (x_{k+1}, x_{k+2}, \ldots, x_{k+l})$.
4 Then, construct another list $\mathscr{L}_2$ sorted according to $\phi(\mathbf{x})$, containing all vectors $\mathbf{x} = (0\mathbf{u})\mathbf{G}_{\text{sys}}$, where $\mathbf{u}$ run through all p-weight vectors of length $k/2$.
5 Add all pairs of vectors $\mathbf{x} \in \mathscr{L}_1$ and $x' \in \mathscr{L}_2$ for which $\phi(\mathbf{x}) = \phi(x')$ and put in a new list $\mathscr{L}$.
6 **if** there exists $\mathbf{x} \in \mathscr{L}$, s.t. $w_{\text{H}}(\mathbf{x}) = t - 2p$ **then**
7 return the codeword $\mathbf{c} \in \mathscr{C}$ corresponding to $\mathbf{x}$.
8 **else**
9 go back to Step 1;
10 **end**

Algorithm 1: Stern's ISD algorithm.

$$(n-k)^2(n+k) + \left(\frac{k}{2} - p + 1\right) + 2l\binom{\frac{k}{2}}{p} + 4p(t-2p+1)\binom{\frac{k}{2}}{p}^2 2^{-l}. \tag{23}$$

We refer to the improved version of ISD attack algorithm in [26, 39], which is a generalization of Stern's algorithm [38]. And they pointed out that for small fields (e.g., in our case, $\mathbb{F}_2$), choosing the new algorithm parameters c and r $(1 < r \le c)$ should be taken into account, which can yield good speedups on the Gaussian elimination cost of each iteration. Furthermore, they offer an improved tool, which allows to compute a rough approximation of the security level of a code-based cryptosystem against information set decoding attacks.

For a practical evaluation of the ISD running times and corresponding security level, similarly, most of the NIST PQCrypto code-based submissions exploited this complexity computation tool to determine the security level of their proposals, and we also use this tool to indicate the security level of our implementation. Note that the ciphertext length should be $n + w$ instead of n in our case. According to this computation tool, we test different input parameters to classify expected bit security level $\kappa := 128, 192, 256$, respectively (see Section 4.6 for more details).

4.5. The KEM of polarRLCE Scheme. Key-encapsulation mechanisms (KEMs) are a common stepping stone aiming for the strong security goal, i.e., indistinguishability against adaptive chosen-ciphertext attacks (IND-CCA2). We also suggest a key-encapsulation mechanism (KEM) version of our polarRLCE scheme, consisting of three algorithms: KEM = (KeyGen, Encaps, Decaps), by applying a transformation of Eaton et al.'s [29] observation. A favorable feature of this proposal is that the process of polarRLCE is convenient, and it enables our KEM-DEM version to achieve the negligible decryption failure rate within the corresponding security parameters. Let $\mathscr{G}$, $\mathscr{H}$, and $\mathscr{K}$ be hash functions, typically SHA-3 as advised by NIST. Here, we show the KEM-DEM version below.

KeyGen (pk, sk): exactly the same as polarRLCE key generation (Section 3), and generate a public/secret key pair (pk, sk).

Encaps ($\mathbf{c}$, K): encapsulate a shared key K in ciphertext $\mathbf{c}$ as follows:

(i) Pick a seed $\mathbf{s} \in \{0,1\}^k$ and set parallel degree P
(ii) For $i \in \{1, 2, \ldots, P\}$, let $\mathbf{e}_i = \mathscr{G}(\mathbf{s} \mid i)$
(iii) Compute $\mathbf{x}_i = \mathbf{s} + \mathscr{H}(\mathbf{e}_i \mid i)$, and the corresponding ciphertext $\mathbf{c}_i = \text{Enc.}(\text{pk}, \mathbf{x}_i, \mathbf{e}_i)$
(iv) Output the shared key $\text{K} = \mathscr{K}(\mathbf{s})$ and $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_P)$

Decaps (K): decapsulate the shared key K from ciphertext $\mathbf{c}$ with sk.

(i) Decrypt ciphertext $\mathbf{c}$ to get $(\mathbf{x}_i^*, \mathbf{e}_i^*)$, where $i \in \{1, 2, \ldots, P\}$.
(ii) Let $j = i$ when the last step successfully decrypt for the current status, then compute $\mathbf{s} = \mathbf{x}_j + \mathscr{H}(\mathbf{e}_j | j)$.
(iii) Compute $\mathbf{c}_i^* = \text{Enc}\cdot(\text{pk}, \mathbf{x}_i^*, \mathbf{e}_i^*)$, obtain $\mathbf{c}^*$ and verify that $\mathbf{c}^* = \mathbf{c}$. If so, return the shared key $\text{K} = \mathscr{K}(\mathbf{s})$.

The same construction was proven to have IND-CCA2 guarantees in the work by [29]. More details regarding the security reduction can be found in [29].

To provide IND-CCA2 for a given security level 2^κ, it is required for the decapsulation to have a correctness error $\delta \le 2^{-\kappa}$. Recall that the DFR of our polarRLCE scheme is no more than 2^{-14}. The resulting ciphertext includes several independent encapsulations with the same key so that a decapsulation failure occurs only if a decryption failure occurs in every instance of the underlying polarRLCE scheme. Willing to target a DFR of $2^{-\kappa}$, we can select the parallel degree $P = 10, 14, 19$, respectively. Thus, our KEM

Table 1: Set of parameters for polarRLCE scheme.

κ	$[n, k, t]$	w	$\mathcal{PK}$	$\mathcal{SK}$	$\mathcal{CT}$	$\mathcal{W}_{\text{ISD}}$
128	$[2^{11}, 500, 285]$	50	97.53	30.54	262.25	130.62
192	$[2^{12}, 585, 760]$	75	256.08	41.81	531.38	193.84
256	$[2^{12}, 960, 610]$	100	379.22	112.55	524.50	257.35

Table 2: Public-key size comparison (in KBytes).

κ	Our	McEliece	RLCE	NTS-KEM	Classic-McEliece
128	97.53	187.69	187.53	312	255
192	256.08	489.4	446.36	907.97	511.88
256	379.22	936.02	1212.86	1386.43	1326

version achieves the desired negligible target DFR value 2^{-140}, 2^{-196}, and 2^{-266}.

4.6. Suggested Parameters for polarRLCE. In this section, we give the suggested parameters in 1 for our polarRLCE scheme, with the three most relevant standard security levels, 128-bit, 192-bit, and 256-bit. Besides, a comparison of the public key size for our suggested parameters with RLCE [19] (the secure second group parameters) and the original McEliece [26] scheme (under binary Goppa codes) is given in 2, together with the state-of-the-art NTS-KEM [41] and Classic-McEliece [42], which are moving on to the 2nd round of the NIST PQC standardization process.

For convenience, we introduce the following notations of each column list in the tables:

(i) κ: security level

(ii) w: the number of inserted columns

(iii) $[n, k, t]$: code length n, code dimension k, and t is the error-correcting ability

(iv) $\mathcal{PK}$: public-key size in kB

(v) $\mathcal{SK}$: private-key size in KBytes

(vi) $\mathcal{CT}$: ciphertext size in Bytes

(vii) $\mathcal{W}_{\text{ISD}}$: the work factor of ISD attack algorithm

We remark that the parameters given in Table 1 may be vulnerable to the attack under the quantum random oracle model (*QROM* [43]). Here, we present the parameters solely to illustrate the rationality of our construction which, to our best knowledge, are secure against current known attacks.

From Table 2, we can see that our scheme can reduce the public key size of the original McEliece scheme by at least 52%. It has the apparent advantage to decrease the key size, especially on the high-security level. However, compared to the candidates based on hamming (rank) quasi-cyclic (QC) codes, the public key size of our proposal is inferior to them. Nevertheless, a new type of statistical analysis, called reaction attacks [44, 45], are threatening these schemes with a specific QC structure of the underlying codes [46, 47]. As a final remark, it would be required to consider the impact of reaction attacks even without the QC structure in our proposal.

5. Conclusion

To summarize, we have proposed a new variant of the code-based encryption scheme by exploring polar codes, benefitting the lower encoding and decoding complexity. We show that, for the proper choice of parameters together with the state-of-the-art cryptanalysis, it is still possible to design secure schemes to achieve the intended security level while keeping a reasonably small key size, using polar code.

However, the disadvantage though is that the information rate is low. We can solve this issue by putting some information data in the error pattern, as shown by Biswas and Sendrier [48]. That is, some additional information bits are mapped into an error vector to be added in the encryption phase. Furthermore, future work in attempting to transfer our scheme to obtain a signature scheme may also be an interesting problem.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We would like to thank Dr. Vlad Dragoi for insightful discussions. This work was supported in part by the National Natural Science Foundation of China (Grant no. 61702124), Qatar Foundation (Grant no. NPRP8-2158-1-423), and Guang Dong Provincial Natural Science Foundation (Grant no. 2018A030310071).

References

[1] P. W. Shor, "Polynomial time algorithms for discrete logarithms and factoring on a quantum computer," in *Lecture Notes in Computer Science*, p. 289, Springer, Berlin, Germany, 1994.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing-STOC*, ACM Press, Philadelphia, PA, USA, May 1996.

[3] NIST, *Post Quantum Crypto Project (2017)*, NIST, Gaithersburg, MD, USA, 2017.

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

[5] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems Control Inform Theory*, vol. 15, no. 2, pp. 159–166, 1986.

[6] H. Janwa and O. Moreno, "Mceliece public key cryptosystems using algebraic-geometric codes," *Designs, Codes and Cryptography*, vol. 8, no. 3, 1996.

[7] T. P. Berger and P. Loidreau, "How to mask the structure of codes for a cryptographic use," *Designs, Codes and Cryptography*, vol. 35, no. 1, pp. 63–79, 2005.

[8] C. Wieschebrink, "Two NP-complete problems in coding theory with an application in code based cryptography," in

Proceedings of the 2006 IEEE International Symposium on Information Theory, IEEE, Seattle, WA, USA, July 2006.

[9] C. Monico, J. Rosenthal, and A. Shokrollahi, "Using low density parity check codes in the McEliece cryptosystem," in *Proceedings of the 2000 IEEE International Symposium on Information Theory (Cat. No. 00CH37060)*, IEEE, Sorrento, Italy, June 2000.

[10] P. Gaborit, "Shorter keys for code based cryptography," in *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pp. 81–91, Bergen, Norway, March 2005.

[11] V. M. Sidelnikov, "A public-key cryptosystem based on binary reed-muller codes," *Discrete Mathematics and Applications*, vol. 4, no. 3, 1994.

[12] P. Gaborit, G. Murat, O. Ruatta, and G. Zémor, "Low rank parity check codes and their application to cryptography," in *Proceedings of the Workshop on Coding and Cryptography WCC*, vol. 2013, Bergen, Norway, April 2013.

[13] M. Baldi and G. F. Chiaraluce, "Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes," in *Proceedings of the 2007 IEEE International Symposium on Information Theory*, IEEE, Nice, France, June 2007.

[14] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich, "Distinguisher-based attacks on public-key cryptosystems using reed-solomon codes," *Designs, Codes and Cryptography*, vol. 73, no. 2, pp. 641–666, 2014.

[15] A. Couvreur, C. I. Marquez, and R. Pellikaan, "A polynomial time attack against algebraic geometry code based public key cryptosystems," in *Proceedings of the 2014 IEEE International Symposium on Information Theory*, IEEE, Honolulu, HI, USA, June 2014.

[16] J. C. Faugere, A. Otmani, L. Perret, and J. P. Tillich, "Algebraic cryptanalysis of McEliece variants with compact keys," in *Advances in Cryptology–EUROCRYPT 2010*, pp. 279–298, Springer, Berlin, Germany, 2010.

[17] L. Minder and A. Shokrollahi, "Cryptanalysis of the sidelnikov cryptosystem," in *Advances in Cryptology-EUROCRYPT 2007*, pp. 347–360, Springer, Berlin, Germany, 2007.

[18] Y. Wang, "Quantum resistant random linear code based public key encryption scheme RLCE," in *Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT)*, IEEE, Barcelona, Spain, July 2016.

[19] Y. Wang, *Rlce-key Encapsulation Mechanism*, NIST, Gaithersburg, MD, USA, 2017.

[20] A. Couvreur, M. Lequesne, and J. P. Tillich, "Recovering short secret keys of RLCE in polynomial time," in *Post-quantum Cryptography*, pp. 133–152, Springer International Publishing, Berlin, Germany, 2019.

[21] E. Arikan, "Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[22] R. Hooshmand, M. K. Shooshtari, T. Eghlidos, and M. R. Aref, "Reducing the key length of mceliece cryptosystem using polar codes," in *Proceedings of the 2014 11th International ISC Conference on Information Security and Cryptology*, IEEE, Tehran, Iran, September 2014.

[23] S. R. Shrestha and Y. S. Kim, "New McEliece cryptosystem based on polar codes as a candidate for post-quantum cryptography," in *Proceedings of the 2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, IEEE, Incheon, Korea, September 2014.

[24] M. Bardet, J. Chaulet, V. Drăgoi, A. Otmani, and J. P. Tillich, "Cryptanalysis of the McEliece public key cryptosystem based on polar codes," in *Post-quantum Cryptography*, pp. 118–143, Springer International Publishing, Berlin, Germany, 2016.

[25] Y. Li, R. H. Deng, and X. Wang, "On the equivalence of mceliece's and niederreiter's public-key cryptosystems," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 271–273, 1994.

[26] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-quantum Cryptography*, pp. 31–46, Springer, Berlin, Germany, 2008.

[27] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Performance limits and practical decoding of interleaved reed-solomon polar concatenated codes," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1406–1417, 2014.

[28] V. Dragoi, *Algebraic approach for the study of algorithmic problems coming from cryptography and the theory of error correcting codes*, Ph.D. thesis, University of Rouen, Mont-Saint-Aignan, France, 2017.

[29] E. Eaton, M. Lequesne, A. Parent, and N. Sendrier, "QC-MDPC: a timing attack and a CCA2 KEM," in *Post-quantum Cryptography*, pp. 47–76, Springer International Publishing, Berlin, Germany, 2018.

[30] I. Cascudo, R. Cramer, D. Mirandola, and G. Zemor, "Squares of random linear codes," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1159–1173, 2015.

[31] C. T. Gueye and E. H. M. Mboup, "Secure cryptographic scheme based on modified reed muller codes," *International Journal of Security and Its Applications*, vol. 7, no. 3, pp. 55–64, 2013.

[32] A. Otmani and H. T. Kalachi, "Square code attack on a modified sidelnikov cryptosystem," in *Lecture Notes in Computer Science*, pp. 173–183, Springer International Publishing, Berlin, Germany, 2015.

[33] C. Wieschebrink, "Cryptanalysis of the niederreiter public key scheme based on GRS subcodes," in *Post-quantum Cryptography*, pp. 61–72, Springer, Berlin, Germany, 2010.

[34] V. Drăgoi, V. Beiu, and D. Bucerzan, "Vulnerabilities of the McEliece variants based on polar codes," in *Innovative Security Solutions for Information Technology and Communications*, pp. 376–390, Springer International Publishing, Berlin, Germany, 2019.

[35] E. Petrank and R. M. Roth, "Is code equivalence easy to decide?," *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1602–1604, 1997.

[36] N. Sendrier, "Finding the permutation between equivalent linear codes: the support splitting algorithm," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1193–1203, 2000.

[37] E. Prange, "The use of information sets in decoding cyclic codes," *IEEE Transactions on Information Theory*, vol. 8, no. 5, pp. 5–9, 1962.

[38] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, pp. 106–113, Springer-Verlag, Berlin, Germany, 1989.

[39] C. Peters, "Information-set decoding for linear codes over f q," in *Post-quantum Cryptography*, pp. 81–94, Springer, Berlin, Germany, 2010.

[40] A. May and I. Ozerov, "On computing nearest neighbors with applications to decoding of binary linear codes," in *Advances in Cryptology–EUROCRYPT 2015*, pp. 203–228, Springer, Berlin, Germany, 2015.

[41] M. Albrecht, C. Cid, K. G. Paterson, C. J. Tjhai, and M. Tomlinson, *NTS-KEM*, NIST, Gaithersburg, MA, USA, 2019.

[42] D. J. Bernstein, T. Chou, T. Lange et al., *Classic Mceliece: Conservative Code-Based Cryptography*, NIST, Gaithersburg, MA, USA, 2019.

[43] D. Boneh, O. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 41–69, Springer, Berlin, Germany, 2011.

[44] Q. Guo, T. Johansson, and P. Stankovski, "A key recovery attack on mdpc with cca security using decoding errors," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 789–815, Springer, Berlin, Germany, 2016.

[45] A. Nilsson, T. Johansson, and P. S. Wagner, "Error amplification in code-based cryptography," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 1, pp. 238–258, 2019.

[46] N. Aragon, P. Barreto, S. Bettaieb et al., *Bike: Bit Flipping Key Encapsulation*, NIST, Gaithersburg, MD, USA, 2019.

[47] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, *Ledacrypt*, NIST, Gaithersburg, MD, USA, 2019.

[48] B. Biswas and N. Sendrier, "McEliece cryptosystem implementation: theory and practice," in *Post-quantum Cryptography*, pp. 47–62, Springer, Berlin, Germany, 2008.