A Novel Polar Code-Based Key Encapsulation Mechanism with Non-permutation Equivalent Public Key

Huiling Zhang[1], Zhiqiang Lin[1], Jingang Liu[2,3(✉)], and Haixiong Zhou[4]

[1] School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, People's Republic of China
[2] School of Mathematics and Systems Science, Guangdong Normal University of Technology, Guangzhou 510665, People's Republic of China
liujingang@gpnu.edu.cn
[3] Henan Key Laboratory of Network Cryptography, Zhengzhou 450001, People's Republic of China
[4] Guangzhou Chinagdn Security Technology Co., Ltd., Guangzhou 510640, People's Republic of China

Abstract. Code-based cryptography is one of the post-quantum cryptography techniques which is able to resist attacks from quantum computers. This paper proposes a novel key encapsulation mechanism (KEM) based on polar codes. As the basic technology of 5G communication, polar codes have efficient encoding and decoding procedures, thus can improve the efficiency of a code-based cryptosystem. We apply polar codes to a variant of the McEliece public-key encryption scheme in which the codes of the public key and secret key are non-permutation equivalent. Then we construct the KEM protocol by Fujisaki-Okamoto transformation method. This KEM is indistinguishably secure from a chosen ciphertext attack. The public key size of the proposed KEM is smaller than that of the Classic McEliece KEM in NIST PQC standardization process, under the same security level.

Keywords: Public-key cryptography · Post-quantum cryptography · Polar code · Key encapsulation mechanism

1 Introduction

Cryptography is an essential foundation of information security technology. However, with the rapid development of quantum computers, public-key cryptosystems such as RSA, ElGamal and ECC, have suffered from potentially catastrophic attacks. This fact impels cryptographic researchers to find new cryptographic primitives which can resist quantum attacks, i.e., the post-quantum cryptography (PQC) [5,19,23]. In December 2016, the National Institute of Standards and Technology (NIST) has launched a PQC standardization project [9],

J. Cai et al. (Eds.): SPNCE 2023, LNICST 525, pp. 188–201, 2025.
https://doi.org/10.1007/978-3-031-73699-5_13

in order to call for cryptosystems that are secure for both quantum and classical computers, and compatible with existing communication protocols and networks. The scope of the cryptosystems includes key-encapsulation mechanisms, public-key encryption schemes, and digital signature schemes. After three rounds of rigorous selection, four standard algorithms were chosen: CRYSTALS-Kyber, CRYSTALS-Dilithium, Falcon, and SPHINCS+ [1,7]. In addition, there are four alternative algorithms that have been preserved: BIKE, Classic-McEliece, and HQC, SIKE. In Europe, as early as 2015, the PQCRYPTO project has been launched and has put forward relevant standardization proposals [14]. In 2018, Chinese Association for Cryptologic Research (CACR) initiated a national cryptographic algorithm design competition. The submitted algorithms are encouraged to be PQC algorithms. Unsurprisingly, the winning public key algorithms that were eventually announced in January 2020 are all quantum-resistant [8].

Key encapsulation mechanism (KEM) is a protocol that uses asymmetric cryptographic algorithms to realize secure exchange of the symmetric keys between two parties of a session. KEM is able to improve the problem of limiting the space of plaintexts in the process of using public key encryption. It is one of the effective ways to solve the problems of key distribution and key management in large-scale networks. Code-based KEM is an important part of the post-quantum cryptography. In 1978, Berlekamp et al. proved that the decoding problem of a general linear code is NP-complete. In the same year, Robert J. McEliece utilized this difficult problem and proposed the McEliece public-key cryptosystem [16]. It is based on Goppa codes, with very fast speed of encryption and decryption. During more than four decades, this scheme has withstood various cryptographic attacks and analyses, and no serious security vulnerabilities have been found. However, the main defect is that its public key size is too large comparing with RSA or ECC, and has not been widely used. For PQC's demand, Classic-McEliece, a code-based KEM which adopts the McEliece public-key cryptosystem as its basic framework [12], has entered the 4th round of the NIST PQC standardization project. In addition, two candidate schemes of the 4th round, BIKE and HQC, are also code-based KEMs. The construction of the BIKE scheme is based on the proposed cyclic medium-density parity-check codes, while the construction of the HQC scheme is based on the problem of decoding difficulty of random cyclic codes on the Hamming metric.

In 2009, Arikan discovered polar codes [2], which is the only code currently available that can reach the Shannon capacity. Polar codes use a low-complexity decoding algorithm, Arikan's successive elimination decoder, to obtain the capacity of any symmetric binary discrete memoryless channel (B-DMC). They have better decoding and error correction capabilities than the Goppa codes used by Classic-McEliece, thus can be applied to reduce the size of the public key. Moreover, since polar codes are the basic codes for 5G wireless communication systems with good performance and low complexity, the polar code-based cryptosystems can be highly integrated with the booming 5G network communication systems.

In 2014, two variants of the McEliece public-key encryption scheme based on polar codes were presented [13,20]. In 2016, Bardet et al. proposed a structural

attack by using minimum weight codewords [4]. This attack attempts to solve the code equivalence problem for polar codes, which may have an impact on the security of this class of cryptosystems. In [15] Liu et al. combine the idea of RLCE scheme [21] and introduce a first polar code-based KEM. By inserting random columns to the secret code generator matrix, this KEM scheme can resist against the code equivalent attacks. In 2022, Reza et al. follow the framework of [13] and present a KEM based on polar codes, called KEM-PC [12]. They avoid the code equivalent attacks by exploiting a special kind of random subcodes of polar codes instead of the original form in [12].

Our Contribution. In this paper, we propose a novel polar code-based KEM which can be immune to code equivalence attacks. The fundamental public-key encryption of the KEM scheme is a variant of McEliece cryptosystem proposed by Baldi et al. [3]. It exhibits the property that the public-key codes are non-permutation equivalent to the code used as the secret codes. We apply the Fujisaki-Okamoto transformation under the random oracle machine model and construct the KEM protocol. Comparing with the Classic-McEliece KEM in the NIST PQC standardization project, our scheme has the following advantages:

Firstly, the generator matrix of the polar codes can be more flexibly selected, with a wider range of choices, which helps to avoid possible attacks against the fixed generator matrix in a code-based cryptosystem. Secondly, the lower encoding and decoding complexity ($\mathcal{O}(nlogn)$) of the polar codes can increase the speed of the system. Thirdly, the non-permutation property of the encryption process is able to the resistance to the code equivalence attacks.

2 Preliminaries

In this section, we briefly introduce some basic background of polar codes and key-encapsulation mechanism. We denote vectors by lower-case bold letters and matrices by upper-case bold letters throughout this paper, e.g., vector $\mathbf{m}$ and matrix $\mathbf{A}$.

2.1 Polar Codes

Polar codes are a class of linear codes proposed by E. Arikan [2] in 2009 based on the phenomenon of channel polar. They are the first channel codes that can be proved theoretically to reach the capacity of any binary input discrete memoryless symmetric channel, and have low encoding and decoding complexity and determinacy.

Suppose $\{W : X \to Y\}$ is a channel with input $X = \{0,1\}^n$ and output $Y = \{0,1\}^n$, respectively. The transition probability of the channel W is defined by $\{W(y|x), x \in X, y \in Y\}$. Polar codes have two important parameters, i.e., $I(W)$ and $Z(W)$. $I(W) \in [0,1]$ is used as a measure of the symmetric mutual information among input and output of channel W, when W is a binary memoryless symmetric (BMS) channel. $Z(W) \in [0,1]$ is known as the Bhattacharyya

parameter of the channel W, which can be used as a criterion for reliability measure. The computational formulas for $I(W)$ and $Z(W)$ is as follows:

$$I(W) \triangleq \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{2W(y|x)}{W(y|0) + W(y|1)},$$

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}.$$

If W is a binary erasure channel with determination probability ε, denoted by BEC(ε), then we have $Z(W) = \varepsilon$ and $I(W) = 1 - Z(W) = 1 - \varepsilon$. When the value of the channel $Z(W) \in [0, 1]$ is smaller, it indicates that this channel is more reliable. The basic idea of polar codes is to transmit data only in channels where $Z(W)$ is close to 0.

Polar codes essentially are a class of binary linear codes, and thus can be encoded by a generator matrix: $\mathbf{x}^n = \mathbf{u}^n \mathbf{G}^n$, where the generator matrix is $\mathbf{G}_n = \mathbf{B}_n \mathbf{F}^{\otimes}$, and $\mathbf{B}_n$ is a sorting matrix to accomplish the inverse order bit by bit. $\mathbf{F}^{\otimes}$ denotes that the matrix $\mathbf{F}$ undergoes n times the operation of Kronecker product and the recursion is:

$$\mathbf{F}^{\otimes 1} = \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$$\mathbf{F}^{\otimes n} = \mathbf{F} \otimes \mathbf{F}^{\otimes n-1}.$$

The main process of polar code encoding can be briefly summarized as follows: 1) reliability estimation, 2) bit mixing, 3) construct the generation matrix, 4) matrix multiplication calculation, 5) output coding. Serial offset decoding based on log-likelihood ratio (LLR) can be used for polar code decoding because LLR is a sufficient statistic of the received signal in the binary input channel, which is numerically stable, and LLR decoding [2] is also used in practical systems. In addition, all decoding algorithms for polar codes have a certain bit error ratio. It will impact on the decryption of the KEM. We will discuss the decryption failure in detail in Subsect. 3.1.

2.2 Key Encapsulation Mechanism

A key encapsulation mechanism consists of four probabilistic polynomial time algorithms, i.e., KEM = (Setup, KeyGen, Encaps, Dncaps), as follows:

- KEM.Setup(λ) $\rightarrow$ pp: A trusted authority runs the Setup algorithm, which takes as input a security parameter λ and outputs the global public parameters pp.
- KEM.KeyGen(pp) $\rightarrow$ $(\mathbf{pk}, \mathbf{sk})$: The user runs the key generation algorithm, which outputs a key pair $(\mathbf{pk}, \mathbf{sk})$ with the public parameters pp as input. The public key $\mathbf{pk}$ is released while the private key $\mathbf{sk}$ is kept secret to the user.

- KEM.Encaps$(pp, \mathbf{pk}) \rightarrow (\mathbf{c}, \mathbf{k})$: An encapsulator runs the encapsulation algorithm and outputs a ciphertext $\mathbf{c} \in \mathcal{C}$ and a key $\mathbf{k} \in \mathcal{K}$ using the public key $\mathbf{pk}$ and the public parameters pp as inputs. The ciphertext $\mathbf{c}$ is publicly broadcast and the encapsulation key $\mathbf{k}$ is kept secret to the encapsulator. Here $\mathcal{C}$ is the ciphertext space and $\mathcal{K}$ is the key space.
- KEM.Decaps$(pp, \mathbf{sk}, \mathbf{c}) \rightarrow (\mathbf{k} \vee \perp)$: A decapsulator runs the decapsulation algorithm with the key $\mathbf{sk}$, the ciphertext $\mathbf{c}$ and the public parameters pp as input and the key $\mathbf{k}$ or $\perp$ as output, Where $\perp$ is the symbol for decapsulation failure.

If for any security parameters λ, pp, $(\mathbf{pk}, \mathbf{sk})$ and $(\mathbf{c}, \mathbf{k})$, the probability of decapsulation failure is satisfied:

$$Pr[KEM.Decaps(pp, \mathbf{sk}, \mathbf{c}) \neq \mathbf{k}] \leq \delta,$$

then the KEM is said to be δ-correct. The KEM is correct if $\delta = 0$.

2.3 IND-CCA Security for Key Encapsulation Mechanisms

A key encapsulation mechanism where an attacker cannot obtain any information about the key even if he has access to the decryption mechanism to decrypt any ciphertext of his choice, which is said to be indistinguishably secure from a chosen ciphertext attack (IND-CCA).

Let $\mathcal{A}$ be an adversary and let Π = (Gen,Encaps,Decaps) be a KEM with key length k. Consider the following IND-CCA experiment $\mathrm{KEM}^{ind-cca}_{\mathcal{A},\Pi}(\lambda)$ and the experiment $\mathrm{KEM}^{ind-cca}_{\mathcal{A},\Pi}(\lambda)$ is visualized as in Fig. 1.

- $\mathrm{Gen}(1^\lambda)$ is run to get the key pair $(\mathbf{pk}, \mathbf{sk})$, and $\mathrm{Encaps}_{\mathbf{pk}}(1^\lambda)$ is run to generate $(\mathbf{k}, \mathbf{c})$.
- Choose a uniform bit $b \in \{0, 1\}$. If $b = 0$ set $\hat{\mathbf{k}} = \mathbf{k}$. If $b = 1$, then choose a uniform $\hat{\mathbf{k}} \in \{0, 1\}^k$.
- $\mathcal{A}$ is given $(\mathbf{pk}, \mathbf{c}, \hat{\mathbf{k}})$ and access to an oracle $\mathcal{O}_{Dcaps(\cdot)}$, but may not request decapsulation of $\mathbf{c}$ itself.
- $\mathcal{A}$ outputs a bit b'. The output of the experiment $\mathrm{KEM}^{ind-cca}_{\mathcal{A},\Pi}(\lambda)$ is defined to be 1 if $b' = b$; otherwise, it is defined to be 0.

Definition 1. *A KEM Π is IND-CCA secure if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there is negligible function negl such that*

$$Pr[KEM^{ind-cca}_{\mathcal{A},\Pi}(\lambda) = 1] \leq \tfrac{1}{2} + negl(\lambda).$$

3 Polar Codes Based Key Encapsulation Mechanism

In this section, we first construct a public key algorithm based on polar codes, then the KEM is obtained by using the Fujisaki-Okamoto transformation.

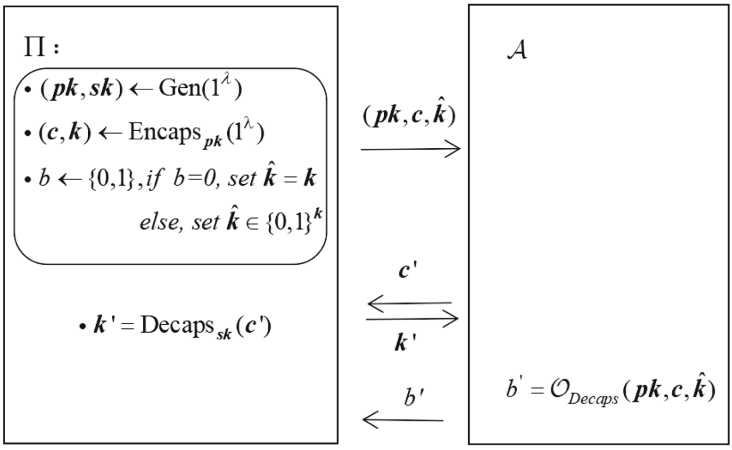

Fig. 1. Flowchart of experimental $\mathrm{KEM}_{\mathcal{A},\Pi}^{ind-cca}(\lambda)$

3.1 A Variant of McEliece Public Key Cryptosystem Based on Polar Codes

In 2016, Baldi proposed a variant of McEliece cryptosystem, which replaces the permutation matrix in the public key with a dense matrix, thus the relationship between the public and private keys no longer permutational equivalent [3]. This scheme can resist to the code equivalence attacks. We apply polar code to this public-key scheme, and the construction is as follows.

- **Key Generation**

Choose an $[n, k]$ polar code, construct a $k \times n$ generating matrix $\mathbf{G}$, and then randomly choose a $k \times k$ non-singular matrix $\mathbf{S}$ and an $n \times n$ dense transformation matrix $\mathbf{Q}$. Let the public key be $\mathbf{G}_{pub} = \mathbf{S^{-1}GQ^{-1}}$ and $\mathbf{a}$, and the private key be $(\mathbf{S}, \mathbf{G}, \mathbf{Q})$, where $\mathbf{Q} = \mathbf{R} + \mathbf{T}$ and $\mathbf{a}$ are constructed as follows:

The matrix $\mathbf{R}$ is an $n \times n$ dense matrix, such that $\mathbf{a}_i, \mathbf{b}_i (1 \leq i \leq 2)$ are matrices defined on $\mathbb{F}_2$ whose size are $1 \times n$. In particular, define $\mathbf{a} = \mathbf{a}_1 + \mathbf{a}_2$ and the matrix $\mathbf{R}$ is denoted by

$$\mathbf{R} = \begin{pmatrix} \mathbf{a} \\ \mathbf{0} \end{pmatrix}^T \cdot \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}.$$

The matrix $\mathbf{T}$ is an $n \times n$ non-singular matrix with average row-column weights $\mu (1 < \mu < 2)$.

- **Encryption**

To encrypt a plaintext $\mathbf{m} \in \mathbb{F}_2^k$, in addition to the use of a public key, a randomly selected error vector $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n)$ with Hamming weight $t_{pub} = \lfloor \frac{t}{\mu} \rfloor$ is

required, where μ is the average row-column weight of the matrix $\mathbf{T}$ and t is the error-correcting capacity of the polar codes. On the other hand, $\mathbf{e}$ is required to satisfy $\mathbf{a} \cdot \mathbf{e}^T = \mathbf{0}$. The ciphertext computation procedure is $\mathbf{c} = \mathbf{mG}_{pub} + \mathbf{e}$.

- **Decryption**

In order to decode the ciphertext $\mathbf{c}$, we first calculate $\mathbf{cQ} = \mathbf{mS}^{-1}\mathbf{G} + \mathbf{eQ}$, i.e., $\mathbf{cQ} = \mathbf{mS}^{-1}G + \mathbf{eR} + \mathbf{eT}$. Since $\mathbf{R} = \mathbf{a}^T \cdot \mathbf{b}_1$ and $\mathbf{a} \cdot \mathbf{e}^T = \mathbf{0}$, then $\mathbf{eR} = \mathbf{e} \cdot \mathbf{a}^T \cdot \mathbf{b}_1 = \mathbf{0}$. Also, $\mathcal{W}_H(\mathbf{eT}) \leq t$ since the average weight of the rows and columns of the matrix $\mathbf{T}$ is μ. Thus, $\mathbf{cQ} = \mathbf{mS}^{-1}\mathbf{G} + \mathbf{eT}$. Finally, the plaintext $\mathbf{m}$ can be decoded using the SC decoding algorithm [2] or other decoding algorithms.

Classic-McEliece based on Goppa codes considered to be secure for one-way chosen plaintext attack (OW-CPA) when appropriate parameters are adopted [16]. The security analyses in [3] and Subsect. 3.4 of this paper show that there is no new effective attack for our polar codes based cryptosystem. Hence it can also be considered as OW-CPA security.

3.2 Construction of the Key Encapsulation Mechanism

We show how to use the Fujisaki-Okamoto transformations to construct a polar code based KEM [11].

Let $\mathcal{G}, \mathcal{H}, \mathcal{K}$ and $\mathcal{L}$ be hash functions. The KEM is described by three algorithms: KeyGen for key generation, Encaps for encapsulation and Decaps for decapsulation.

Algorithm 1. KeyGen

Input: Security parameter λ.
Output: Public key **pk** and secret key **sk**.

1: Choose an $[n, k]$ polar code, construct a $k \times n$ generating matrix $\mathbf{G}$.
2: Randomly choose a $k \times k$ non-singular matrix $\mathbf{S}$ and an $n \times n$ dense transformation matrix $\mathbf{Q}$, where $\mathbf{Q} = \mathbf{R} + \mathbf{T}$. (Matrix construction omitted, the same as **Key Generation** 3.1)
3: **return** secret key $\mathbf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{Q})$ and public key $\mathbf{pk} = (\mathbf{S}^{-1}\mathbf{GQ}^{-1}, \mathbf{a})$. (The construction of $\mathbf{a}$ is omitted and is the same as for **Key Generation** 3.1)

Unlike Goppa codes, the decoding algorithm of polar codes exists error probability, which will lead to the decryption failure of the polar code-based public-key cryptosystem. The decapsulation failure rate (DFR) can be estimated by the method of [10], which is depended on the Bhattacharyya parameter $Z(W)$ and the code parameters $[n, k, t]$. The parameters selected in Subsect. 3.4 can ensure that the DFR in our scheme is no more than 2^{-15} according to [10].

To further reduce the probability of the decapsulation failure, Algorithm 4 adopts parallel encryption operations, that is, P error vectors $\mathbf{e}_i$ are generated simultaneously, corresponding to P ciphertexts $\mathbf{c}_i$. The decapsulation can be

Algorithm 2. Encaps

Input: Public key $\mathbf{pk}$ and $\mathbf{s} \leftarrow seed \in \{0,1\}^k$.
Output: The key $\mathbf{k}$ and key encapsulation $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_P)$.

for $i = 1$ to P **do**
  Let $\mathbf{e}_i = \mathcal{G}(\mathbf{s}||i)$.// see algorithm 4.
  Compute $\mathbf{x}_i = \mathbf{s} + \mathcal{H}(\mathbf{e}_i||i)$.
  Compute $\mathbf{c}_i = Enc_{\mathbf{pk}}(\mathbf{x}_i, \mathbf{e}_i)$.
end for
Compute $\mathbf{k} = \mathcal{K}(\mathbf{s})$.
return the key $\mathbf{k}$ and key encapsulation $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_P)$.

Algorithm 3. Decaps

Input: Secret key $\mathbf{sk}$, public key $\mathbf{pk}$, and encapsulation $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_P)$.
Output: The key $\mathbf{k} := \mathcal{K}(\mathbf{s}^*)$ or decapsulation failure " $\perp$ ".

for $i = 1$ to P **do**
  Run $(\mathbf{x}_i^*, \mathbf{e}_i^*) \leftarrow \mathrm{Dec}(\mathbf{sk}, \mathbf{c}_i)$.
  if $\mathrm{Dec}(\mathbf{sk}, \mathbf{c}_i)$ successfully decoded for the first time **then**
    Set used index $j = i$.
    if $\mathrm{Dec}(\mathbf{sk}, \mathbf{c}_i)$ failed to decode for $i = 1$ to P **then**
      Return decapsulation failure " $\perp$ ".
    end if
  end if
  Compute $\mathbf{s}^* = \mathbf{x}_i^* + \mathcal{H}(\mathbf{e}_i^*||i)$.
  Compute $\mathbf{e}_i^{**} = \mathcal{G}(\mathbf{s}^*||i)$, $\mathbf{x}_i^{**} = \mathbf{s}^* + \mathcal{H}(\mathbf{e}_i^{**}||i)$ and $\mathbf{c}_i^{**}$=$\mathrm{Enc}_{\mathbf{pk}}(\mathbf{x}^{**}, \mathbf{e}^{**})$.
end for
if $\mathbf{c}_i = \mathbf{c}_i^{**}$ for all $i = 1, 2, \cdots, P$ **then**
  Return $\mathbf{k} := \mathcal{K}(\mathbf{s}^*)$.
else
  Return decapsulation failure " $\perp$ ".
end if

Algorithm 4. Error vector $\mathbf{e}$ generation

Input: A binary seed vector $\mathbf{s} \leftarrow seed \in \{0,1\}^k$, integers n and $t_{pub} = \lfloor \frac{t}{\mu} \rfloor$.
Output: A binary error vector $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n)$ of length n and weight t_{pub}.

Set $\mathbf{e} \leftarrow 1^{t_{pub}}||0^{n-t_{pub}}$.
for $i = 1$ to P **do**
  $j \leftarrow \mathcal{F}(\mathbf{s}) mod(n-i)$: Truncate $\mathcal{L}(s)$ to a string with q bytes and $q > n$. Convert the string with q bytes to an integer Z.
  if $Z > 2^{8q} - (2^{8q} mod(n-i))$ **then**
    Go back step 3.
  else
    Return $j = Z mod(n-i)$.
  end if
end for
Switch the $\mathbf{e}_i$ and $\mathbf{e}_{i+j}$ positions in $\mathbf{e}$.
return $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n)$.

completed as long as one of the $\mathbf{c}_i$ is decrypted correctly. Therefore we can control the failure rate to a desired range. For instance, to provide security levels $\mathbf{k} := 128, 192, 256$, we can choose $P = 9, 13, 18$, respectively. Thus, our KEM protocol achieves the desired negligible target DFR value $2^{135}, 2^{195}, 2^{270}$.

3.3 Proof of IND-CCA Secure

We now prove that our proposed polar code-based KEM is IND-CCA secure under the random oracle model.

Theorem 1. *If the public-key scheme described in Subsect. 3.1 is OW-CPA secure, and if $\mathcal{G}, \mathcal{H}, \mathcal{R}$ can be used as a random oracle machine, then the polar code-based key encapsulation mechanism proposed in this section is IND-CCA secure.*

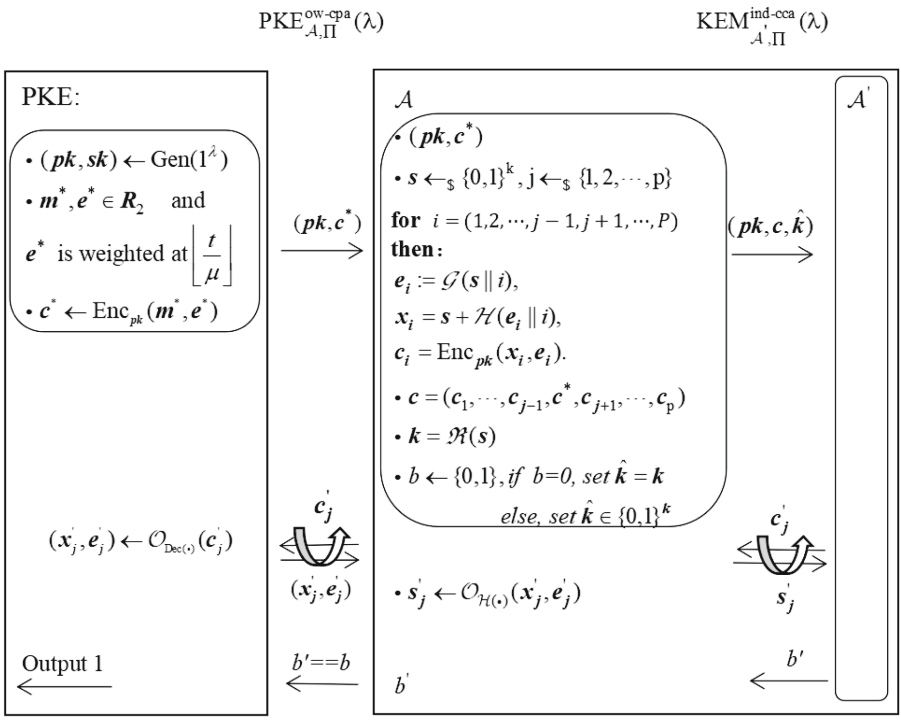

Fig. 2. Visualization of the IND-CCA Security Reduction Experiment

Proof. A visualization of the proof is shown in Fig. 2.

In the $\mathrm{PKE}^{ow-cpa}_{\mathcal{A},\Pi}(\lambda)$ experiment. The public-key encryption scheme generates a key pair $(\mathbf{pk}, \mathbf{sk})$, encrypts the plaintext $\mathbf{m}^*$ to generate the ciphertext $\mathbf{c}^*$,

and sends $(\mathbf{pk}, \mathbf{c}^*)$ to the adversary $\mathcal{A}$, who is essentially acted as a KEM. Adversary $\mathcal{A}$ randomly selects a seed $\mathbf{s}$ of length k, and in turn, for all $i = 1, 2, ..., P$, generates $\mathbf{e}_i$, $\mathbf{x}_i$, and $\mathbf{c}_i$, respectively, by the hash function. A random index j is chosen, and the adversary replaces the position of $\mathbf{c}_j$ with $\mathbf{c}^*$. Finally, the adversary A obtains $\mathbf{c} = (\mathbf{c}_1, \cdots, \mathbf{c}_{j-1}, \mathbf{c}^*, \mathbf{c}_{j+1}, \cdots, \mathbf{c}_P)$ and the key $\mathbf{k} = \mathcal{K}(s)$. Choose a bit $b \in \{0, 1\}$, and if $b = 0$, let $\hat{\mathbf{k}} = \mathbf{k}$; else, let $\hat{\mathbf{k}} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{k}}$. Finally, adversary $\mathcal{A}$ provides $(\mathbf{pk}, \mathbf{c}, \hat{\mathbf{k}})$ to adversary $\mathcal{A}'$.

In the $\mathrm{PKE}^{ind-cca}_{\mathcal{A}',\Pi}(\lambda)$ experiment. Adversary $\mathcal{A}'$ receives $(\mathbf{pk}, \mathbf{c}, \hat{\mathbf{k}})$ and accesses the oracle, which generates a new ciphertext $\mathbf{c}'_j$ using the public key. Then, it sends the newly generated ciphertext $\mathbf{c}'_j$ to the public key encryption algorithm in the $\mathrm{PKE}^{ow-cpa}_{\mathcal{A},\Pi}(\lambda)$ experiment. The decryption oracle responds to $\mathbf{c}'_j$ by returning $(\mathbf{x}'_j, \mathbf{e}'_j)$. Adversary $\mathcal{A}$ accesses the decryption oracle $\mathcal{O}_{\mathcal{H}(\cdot)}$ with $(\mathbf{x}'_j, \mathbf{e}'_j)$ and provides the corresponding seed $\mathbf{s}'_j$, which is then returned to adversary $\mathcal{A}'$. The process of sending $\mathbf{c}'_j$ and receiving the corresponding seed $\mathbf{s}'_j$ can be repeated multiple times. Eventually, adversary $\mathcal{A}'$ outputs a bit b' based on the data comparison. If $b = b'$, the $\mathrm{PKE}^{ow-cpa}_{\mathcal{A},\Pi}(\lambda)$ experiment outputs 1; otherwise, it outputs 0.

Assuming that the constructed key encapsulation mechanism is not IND-CCA secure, there exists a probabilistic polynomial time adversary $\mathcal{A}'$ such that $\Pr[\mathrm{KEM}^{ind-cca}_{\mathcal{A}',\Pi}(\lambda) = 1] = \frac{1}{2} + \varepsilon$, with ε is a non-negligible function. Whereas the public-key encryption scheme is OW-CPA secure, and the probability that a probabilistic polynomial time adversary breaks it is $\Pr[\mathrm{PKE}^{ow-cpa}_{\mathcal{A},\Pi}(\lambda) = 1] = \frac{1}{2} + negl(\lambda)$, where $negl(\lambda)$ is a negligible function. It is evident that the success of the key encapsulation mechanism adversary $\mathcal{A}'$ implies the success of the public-key encryption scheme adversary $\mathcal{A}$ among the experimentally viewable security protocols in Fig. 2. However, the probability of success for $\mathcal{A}'$, which is $\frac{1}{2} + \varepsilon$, is not equal to the probability of success for $\mathcal{A}$, which is $\frac{1}{2} + negl(\lambda)$. Therefore, there is a contradiction, and the assumption is not valid.

3.4 Security Analysis

According to Theorem 1, the security of the polar code-based KEM is totally depended on the public-key cryptosystem in Subsect. 3.1. We will discuss the possible attacks against the public-key cryptosystem, and illustrate that it is OW-CPA secure. Finally, we give the suggested parameters for the scheme.

- **Key recover attacks**

The adversary attempts to obtain the private key $(\mathbf{S}, \mathbf{G}, \mathbf{Q})$ from the public key $\mathbf{G}_{pub} = \mathbf{S}^{-1}\mathbf{G}\mathbf{Q}^{-1}$ by exhaustively guessing $\mathbf{S}$ and $\mathbf{Q}$. The invertible matrix $\mathbf{S}$ has a total of $\Pi_{i=1}^{k}(2^k - 2^{i-1}) > 2^{k^2-2}$ on $\mathbb{F}_2$. As for the private key $\mathbf{Q}$, without considering the structural attack, the adversary can only treat it as a random $n \times n$ matrix, so $\mathbf{Q}$ has a total of $\Pi_{i=1}^{n}(2^n - 2^{i-1}) > 2^{n^2-2}$ on $\mathbb{F}_2$. It is obviously that the brute force attack is invalid.

In the design of the public-key cryptosystem, we disclose $\mathbf{a}$, which is part of the private key $\mathbf{Q}$. This vulnerability is discussed in detail in [3]. The adversary

can first calculate the parity check matrix $\mathbf{H}_{pub}$ of the public key $\mathbf{G}_{pub}$, and let $\mathbf{H}' = (\mathbf{H}_{pub}, \mathbf{a})$. Then according to the constraints $\mathbf{a} \cdot \mathbf{e}^T = \mathbf{0}$, the adversary is able to obtain an equivalent subcode with respect to the private key $\mathbf{G}$, and construct distinguishers through the code equivalent attack [22]. However, the above vulnerability is no longer available when we apply polar codes. Polar codes has a large Hull as well as a group of equivalence codes, thus can avoid the code equivalent attack.

Recently, Bardet et al. introduced a very effective structural attack [20] on the McEliece cryptosystem based on polar codes [4]. They manage to determine the structure of the minimum weight codeword of the original polar codes. Then the code equivalence problem for polar codes with respect to decreasing monomial codes is solved. However, the authors in [4] point out that this attack is very specific and applies only to the original McEliece scheme [20] used polar codes. It is impracticable for our proposed public-key cryptosystem. Notice that the scheme in this paper is of non-permutation, and the generator matrices are of randomly chosen. The codes of the private key are no longer equivalent to the original polar codes. Therefore, our proposed public-key cryptosystem is able to avoid key recovery attacks.

- **Message decoding attacks**

The idea of the message decoding attacks is to recover private messages from ciphertexts. It is an important issue in code-based cryptography. This problem is directly related to the hardness of generic decoding for linear code. One famous message decoding attack is the information set decoding (ISD) algorithm. The ISD algorithm decodes a random $[n, k, t]$ linear code by searching for a number of information sets such that positions are all out of the information sets. It does not depend on any structural properties of the code. Hence, the ISD algorithm is used to measure the choosing of the parameters. If the parameters of the public-key cryptosystem are well selected, the adversary cannot recover the error vector from the ciphertext and public keys, thus the cryptosystem is of OW-CPA secure. The general form of the ISD algorithm is shown as following [18]:

Algorithm 5. General form of the ISD algorithm

Input: Matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, received ciphertext $\mathbf{c} \in \mathbb{F}_2^n$ and weight parameter t.
Output: Message $\mathbf{m} \in \mathbb{F}_2^k$.
1: Select a random subset $\Gamma \in \{1, 2, \cdots, n\}$.
2: Compute $\mathbf{x} = \mathbf{x}_\Gamma \mathbf{G}_\Gamma$.
3: **if** $\mathcal{W}_H(\mathbf{c} + \mathbf{x}\mathbf{G}) \leq t$ **then**
4: Set back $\mathbf{x}$.
5: **else**
6: Return to step 1.
7: **end if**

In this paper, we use an optimized version of the ISD algorithms, i.e., the Stern's algorithm [6,17] to compute a rough approximation of the security level. In fact, most of the NIST PQCRYPTO code-based submissions utilize this complexity computation tool to determine the security level of their proposals.

According to Stern's algorithm, we provide the suggested parameters in Table 1 for our proposed scheme, with the three most relevant standard security levels, $\mathbf{k} := 128, 192, 256$.

Table 1. Suggested Security Parameter Selection.

Security level	$[n, k, t]$	Public Key(KB)	Private Key(KB)	Ciphertext (KB)	ISD Attack Overhead
128	$[2^{10}, 700, 70]$	27.69	59.81	0.125	$\mathcal{O}(2^{133.82})$
192	$[2^{11}, 1400, 110]$	110.74	239.26	0.25	$\mathcal{O}(2^{195.49})$
256	$[2^{12}, 2295, 205]$	504.55	642.95	0.5	$\mathcal{O}(2^{262.47})$

Table 2 lists the public key, private key, and ciphertext parameters of the Goppa-based Classic-McEliece KEM scheme in the NIST PQC standardization project.

Table 2. Classic-McEliece Program Parameters.

Security level	Public Key(KB)	Private Key(KB)	Ciphertext (KB)
128	225	6.3	0.125
192	511.88	13.25	0.19
256	1326	13.75	0.23

Observing from the comparison of Table 1 and Table 2, it can indicate that the size of the ciphertexts are similar between the Classic-McEliece scheme and our scheme. The public key size in our scheme is significantly shorter, which will improve the communication efficiency of the cryptosystem. However, It is also noted that the private key size of our scheme is larger than that of the Classic-McEliece scheme. This is due to the fact that Goppa codes can be generated in polynomial form, while polar codes do not have this property.

4 Conclusion

In this paper, a secure and efficient key encapsulation mechanism is constructed by taking advantage of the qualities of flexible structure and fast decoding speed of polar codes. We first apply polar codes to a variant of the McEliece cryptosystem with non-permutation property, which is able to resist the code equivalent attacks. Then we use the Fujisaki-Okamoto transformation and construct the

polar code-based KEM. To reduce the probability of the decapsulation failure, parallel encryption operations are employed in the proposed KEM, hence we can obtain a negligible security level without changing the parameters.

The IND-CCA secure proof to our proposed KEM is presented under the random oracle model. The proof indicates that the security of the KEM is depended on the OW-CPA secure of the public-key cryptosystem in Subsect. 3.1. We analyze the security of the proposed public-key cryptosystem for the resistance of key recovery attacks and message recovery attacks, and declare it is OW-CPA secure. Finally, we give suggested parameters according to the NIST PQC standardization project standard. Comparing with the Classic-McEliece KEM, our scheme is able to significantly shorten the size of the public key. Furthermore, since polar codes are the basic code of 5G communication, our scheme is expected to be developed into a PQC standard algorithm under the new era of information and communication technology.

Acknowledgements. This work is supported in part by the National Key Research and Development Program of China (Grant No. 2021YFB3100200), Henan Key Laboratory of Network Cryptography Technology (LNCT2021-A01) and Guangzhou basic and applied basic research project (No. 202201011213).

References

1. Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates. https://csrc.nist.gov/News/2022/pqc-candidates-to-be-stand
2. Arikan, E.: Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans. Inf. Theory **55**(7), 3051–3073 (2009)
3. Baldi, M., et al.: Enhanced public key security for the McEliece cryptosystem. J. Cryptol. **29**, 1–27 (2016)
4. Bardet, M., Chaulet, J., Dragoi, V., Otmani, A., Tillich, J.-P.: Cryptanalysis of the McEliece public key cryptosystem based on polar codes. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 118–143. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_9
5. Bernstein, D.J.: Introduction to post-quantum cryptography. In: Post-Quantum Cryptography, pp. 1–14. Springer, Heidelberg (2009)
6. Bernstein, D.J., Lange, T., Peters, C.: Attacking and defending the McEliece cryptosystem. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 31–46. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88403-3_3
7. Boutin, C.: NIST announces first four quantum-resistant cryptographic algorithms. In: National Institute of Standards and Technology (2022)
8. CACR: Chinese national cryptographic algorithm design competition (in Chinese). https://www.cacrnet.org.cn/site/content/854.html
9. Chen, L., et al.: Report on post-quantum cryptography, vol. 12. US Department of Commerce, National Institute of Standards and Technology (2016)
10. Dragoi, V.: Algebraic approach for the study of algorithmic problems coming from cryptography and the theory of error correcting codes. Ph.D. thesis. Université de Rouen, France (2017)

11. Eiichiro et al.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptol. **26**(1), 80–101 (2013)
12. Hooshmand, R., Khoshfekr, M.: Key encapsulation mechanism based on polar codes. IET Commun. **16**, 2438–2447 (2022)
13. Hooshmand, R., et al.: Reducing the key length of McEliece cryptosystem using polar codes. In: 2014 11th International ISC Conference on Information Security and Cryptology, pp. 104–108. IEEE (2014)
14. Lange, T.: PQCRYPTO Project in the EU. In: NIST Workshop on Cybersecurity in a Post-Quantum World (2015)
15. Liu, J., et al.: polarRLCE: a new code-based cryptosystem using polar codes. Secur. Commun. Netw. 2019(2), 1–10 (2019)
16. McEliece, R.J.: A public-key cryptosystem based on algebraic. In: Coding Thv, vol. 4244, pp. 114–116 (1978)
17. Peters, C.: Information-set decoding for linear codes over Fq. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 81–94. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12929-2_7
18. Prange, E.: The use of information sets in decoding cyclic codes. IRE Trans. Inf. Theory **8**(5), 5–9 (1962)
19. Shor, P.W.: Polynomial time algorithms for discrete logarithms and factoring on a quantum computer. In: Adleman, L.M., Huang, M.-D. (eds.) ANTS 1994. LNCS, vol. 877, pp. 289–289. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58691-1_68
20. Shrestha, S.R., Kim, Y.S.: New McEliece cryptosystem based on polar codes as a candidate for post-quantum cryptography. In: International Symposium on Communications & Information Technologies, pp. 368–372 (2014)
21. Wang, Y.: Quantum resistant random linear code based public key encryption scheme RLCE. In: 2016 IEEE International Symposium on Information Theory (ISIT), pp. 2519–2523. IEEE (2016)
22. Wieschebrink, C.: Cryptanalysis of the niederreiter public key scheme based on GRS subcodes. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 61–72. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12929-2_5
23. Zhang, H., et al.: Review on cyberspace security. SCIENTIA SINICA Informationis **46**(2), 125–164 (2016)