BAREKENG: Journal of Mathematics and Its Applications
June 2026 Volume 20 Issue 2 Page 1061–1076
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss2pp1061-1076

MATHEMATICAL ANALYSIS OF QC-MDPC STRUCTURES IN BIKE V5.2 POST-QUANTUM KEY ENCAPSULATION SCHEME

Rosa[1*], Sa'aadah Sajjana Carita[2], Nadia Paramita Retno Adiati[3], Sri Rosdiana[4]

[1,2,3,4]*Department of Cryptographic Engineering, Politeknik Siber dan Sandi Negara*
Jln. Raya H. Usa, Putat Nutug, Kec. Ciseeng, Kabupaten Bogor, Jawa Barat, 16120, Indonesia

*Corresponding author's e-mail: * **rosa@student.poltekssn.ac.id***

Article Info

Article History:

Received: 30th April 2025
Revised: 4th August 2025
Accepted: 23rd September 2025
Available online: 26h January 2026

Keywords:

BIKE v5.2;
QC-MDPC;
Key encapsulation;
Mechanism;
Post-quantum cryptography

ABSTRACT

The security of the BIKE scheme depends on a complex mathematical structure built upon QC-MDPC codes. This scheme is constructed using the Niederreiter framework and the application of $FO^{\not\perp}$ transformation. Its security is based on the complexity of two main mathematical problems: the QCSD Problem and the QCCF Problem. The BIKE v5.2 scheme is the latest version of this scheme. This study aims to mathematically analyze the characteristics forming the BIKE v5.2, focusing on QC-MDPC codes, the Niederreiter framework, and the $FO^{\not\perp}$ transformation, as well as the QCSD and QCCF problems. The method used in this study is a systematic literature review combined with theoretical analysis. The study highlights how the interplay of these three components forms a rational and resilient design. Although the BIKE v5.2 scheme was not selected for standardization by NIST, it is still capable of producing an efficient, secure, and relevant KEM for post-quantum cryptography. Through mathematical analysis of the QC-MDPC construction, the formulation of the complex computational problems QCCF and QCSD, and the rational design of the Niederreiter framework with the $FO^{\not\perp}$ transformation, this study demonstrates that BIKE has a strong security foundation and resistance to both classical and quantum attacks.

How to cite this article:

Rosa and S. S. Carita, N. P. R. Adiati and S. Rosdiana., "MATHEMATICAL ANALYSIS OF QC-MDPC STRUCTURES IN BIKE V5.2 POST-QUANTUM KEY ENCAPSULATION SCHEME", *BAREKENG: J. Math. & App.*, vol. 20, no. 2, pp. 1061-1076, Jun, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Cryptography is a mathematical method used to ensure information security [1]. There are two main categories in cryptography: symmetric and asymmetric. Asymmetric cryptography emerged as a solution to the key distribution problem in symmetric cryptography [2]. One of the approaches developed is the key encapsulation mechanism (KEM), which enables two parties to securely exchange secret keys over a public channel [3]. The security of KEM generally relies on the computational problems of factorization and discrete logarithms [4]. These problems are considered computationally hard and form the basis of algorithms such as RSA, DSA, and ECDSA. Although difficult to solve using classical computation, advancements in quantum computing have shown that these problems can be solved in polynomial time using Shor's algorithm with the assistance of Cryptographically Relevant Quantum Computers (CRQC) [5]. In response to this threat, NIST launched the Post-Quantum Cryptography competition, known as the NIST Call for Proposals, in 2016 to evaluate and establish standardized post-quantum cryptographic algorithms that are resistant to Shor's algorithm when executed on CRQC [6]. One of the evaluated categories was the KEM scheme.

In principle, KEM schemes in the PQC standardization process frequently employ the Fujisaki-Okamoto (FO) framework to strengthen security from Indistinguishability under Chosen-Plaintext Attack (IND-CPA) to Indistinguishability under Chosen-Ciphertext Attack (IND-CCA), while adopting several approaches [4], [6]. One of the earliest and most influential approaches is the code-based paradigm, originating from the McEliece scheme. This scheme utilizes generator matrices and error vectors for encryption, offering strong security despite the drawback of very large public key sizes. Niederreiter subsequently introduced a dual variant of McEliece that employs parity-check matrices, resulting in more compact keys while preserving equivalent security guarantees [4], [7]. These two schemes have become fundamental to code-based cryptography and laid the groundwork for the emergence of modern PQC candidates. The integration of the Niederreiter construction with the FO transformation later gave rise to a new generation of KEM designs. One of these algorithms is the Bit Flipping Key Encapsulation (BIKE) scheme [6].

The BIKE scheme was introduced in 2017 as a candidate for post-quantum cryptography, based on Quasi-Cyclic Moderate-Density Parity-Check (QC-MDPC) codes [7]. BIKE is constructed upon the Niederreiter framework and incorporates an implicit-rejection variant of the Fujisaki-Okamoto ($FO^{\not\perp}$) transformation. The Niederreiter approach is used to perform encryption by generating a ciphertext through the multiplication of a random error vector with a parity-check matrix [8]. Meanwhile, the $FO^{\not\perp}$ transformation is applied to enhance the scheme's security against IND-CCA on the ciphertext [9]. The security of BIKE relies on the complexity of the Quasi-Cyclic Codeword Finding (QCCF) and Quasi-Cyclic Syndrome Decoding (QCSD) problems. These problems are computationally difficult to solve without knowledge of the private key matrix and the intentionally introduced error [10]. However, the BIKE scheme faced challenges during the post-quantum cryptography competition, particularly concerning its high Decoder Failure Rate (DFR) [4]. Following several stages of analysis and evaluation, the BIKE scheme has continued to evolve, culminating in its latest version in 2024: the BIKE v5.2 scheme [7].

As part of a comprehensive evaluation of KEM candidates, NIST conducted a thorough analysis of each remaining algorithm in the fourth round. In the NIST Internal Report NIST IR 8545, titled *"Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process"* released on March 11, 2025, NIST officially concluded the standardization process initiated by the NIST Call for Proposals in 2016. The report announced that the BIKE scheme was not selected for standardization. The primary reason behind NIST's decision was the instability of its Decoding Failure Rate (DFR) analysis. Throughout the evaluation process, BIKE's DFR estimates exhibited uncertainty, raising concerns about the long-term consistency and robustness of the scheme [11].

Nevertheless, BIKE v5.2 remains an interesting and relevant subject of research in the field of post-quantum cryptography. The scheme is built upon QC-MDPC codes, which offer computational efficiency in shared secret generation and produce relatively smaller ciphertext sizes compared to other candidates in the code-based category [11]. Furthermore, to date, no attack has successfully broken the BIKE scheme, either theoretically or practically, within the recommended parameters [12]. Despite research that has explored potential vulnerabilities, such as timing and reaction attacks, the scheme has demonstrated resilience against various exploit attempts [4]. This resilience indicates that BIKE possesses a strong mathematical foundation and remains relevant for continued investigation.

A deeper understanding of the mathematical structure of BIKE v5.2 is essential to clarify the uncertainties regarding its DFR stability and long-term robustness. This study analyzes the structural foundations of BIKE v5.2, focusing on its foundational basis, the QC-MDPC code, as well as its underlying frameworks, namely the Niederreiter construction and the $FO^{\not\perp}$ transformation, to provide a comprehensive mathematical assessment. Through this approach, the study aims to establish a solid foundation to support further evaluations of the effectiveness and robustness of BIKE v5.2, while also contributing to the development of more stable post-quantum cryptographic designs in the future.

2. RESEARCH METHODS

This study employs a systematic literature review (SLR) combined with theoretical analysis to examine the structural characteristics of BIKE v5.2. The SLR was carried out by selecting sources from NIST reports, BIKE specifications, and peer-reviewed journals, with inclusion criteria focusing on the Niederreiter framework, the $FO^{\not\perp}$ transformation, and the QCSD and QCCF problems. The scope of the theoretical analysis covers three dimensions. First, QC-MDPC as the foundational code structure. Second, The Niederreiter framework functions as a mechanism for parameter efficiency. Third, the $FO^{\not\perp}$ transformation plays a crucial role in strengthening security. These findings are then related to the mathematical formulation of QCSD and QCCF, providing a clear rationale for the BIKE v5.2 design. The BIKE v5.2 scheme is illustrated in Fig. 1, adapted from [12].

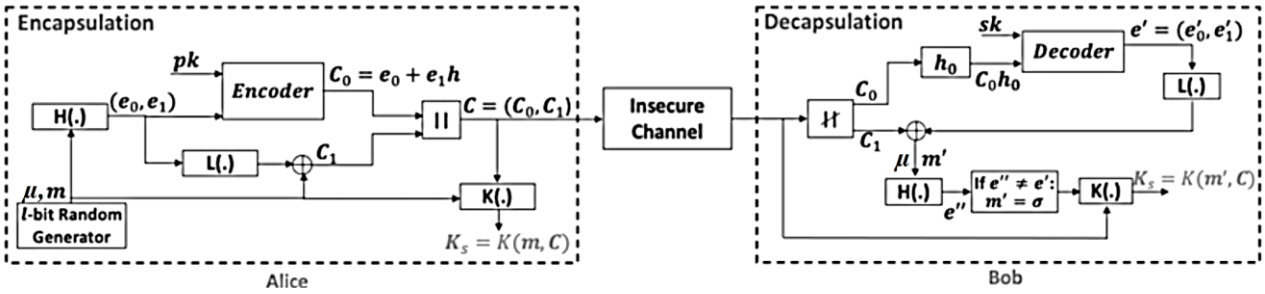

Figure 1. Illustration of the BIKE v5.2 Scheme

The BIKE v5.2 scheme consists of three cryptographic procedures: key generation, encapsulation, and decapsulation, which are respectively described in **Algorithm 1, Algorithm 2** and **Algorithm 3**. Within the encryption system setup, both parties agree on a set of system parameters, as presented in Table 1.

Table 1. Setup

Setup	
Input: Security parameter $\lambda \in \mathbb{Z}^+$	
Output: System parameter Hash functions Decoder	
System Parameters	**Hash Functions**
1. $r \in \mathbb{P}$, with $\text{ord}_r(2) = r - 1$; ensures 2 is a generator mod r.	1. H : $\mathcal{M} \times \mathcal{M} \to \mathcal{E}_t$, using the Fisher-Yates for Constant Weight Words (FY-CWW) algorithm with input 2ℓ.
2. $n = 2r$; defines the code length, where r as circulant block size.	2. K : $\mathcal{M} \times \mathcal{R} \times \mathcal{M} \to \mathcal{K}$, computed using SHA-384, taking the 256 least significant bit (LSB) of the output.
3. $w \in 2\mathbb{Z}^+$, $w \approx \sqrt{n}$, $w/2$ is odd; defines private key weight.	3. L : $\mathcal{R}^2 \to \mathcal{M}$, computed using SHA-384, taking the 256 LSB of the output.
4. $t \in 2\mathbb{Z}^+$; defines error weight.	**Decoder**
5. $l \in \mathbb{Z}^+, 2^\lambda \leq 2^l$; security level λ.	1. Algorithm with DFR $\leq 2^{-\lambda}$ or BIKE-Flip.

The key generation phase in the BIKE scheme produces a private and public key pair that is subsequently used in the encapsulation and decapsulation procedures. This process ensures that only entities in possession of the private key are able to access the encrypted information. The key generation procedure in the BIKE scheme is described in **Algorithm 1**.

Algorithm 1. **BIKE Key Generation**

Input: -
Output: Parameter private key $(h_0, h_1, \sigma) \in \mathcal{H}_w \times \mathcal{M}$
Public key $h \in \mathcal{R}$

1: Generate a private key pair $(h_0(x), h_1(x)) \in \mathcal{H}_w$ using FY-CWW algorithm and (h_0, h_1) denotes the binary representation of private key pair $(h_0(x), h_1(x))$.
2: Compute $h(x) = h_1(x) \cdot h_0^{-1}(x) \bmod (x^r - 1)$ and h denotes the binary representation of $h(x)$.
3: Derive the $\mu = \pi_\ell(h)$, where π_ℓ denotes a function that extracts the ℓ most significant bits of its input.
4: Select $\sigma \in \mathcal{M}$ uniformly at random using the SHAKE256 hash function.

The encapsulation process is executed by the sender who intends to securely transmit information. This procedure is formally presented in **Algorithm 2**.

Algorithm 2. **BIKE Encapsulation**

Input: Public key $h \in \mathcal{R}$
Output: Shared key $K \in \mathcal{K}$
Ciphertext $c \in \mathcal{R} \times \mathcal{M}$

1: Choose a random message $m \in \mathcal{M}$, generated using SHAKE256.
2: Define $\mu = \pi_\ell(h)$ and compute the error pair $(e_0, e_1) = \mathrm{H}(m, \mu)$.
3: Compute the $c = [c_0|c_1]$, where $c_0(x) = e_0(x) + e_1(x) \cdot h(x) \bmod (x^r - 1)$ and c_0 denotes the binary representation of $c_0(x)$. The second component is computed as $c_1 = m \oplus \mathrm{L}(e_0, e_1)$.
4: Derive the shared key $K = \mathrm{K}(m, c)$.

The auxiliary value μ is utilized to prevent multi-target attacks on the public key. A multi-target attack refers to a scenario in which an adversary reuses a fixed error vector pair (e_0, e_1) to generate multiple valid ciphertexts across different public keys h. This vulnerability arises because the error vectors are not intrinsically bound to the public key. To mitigate this, the BIKE v5.2 scheme binds μ to the public key h, effectively ensuring that each error pattern remains unique to a specific key [13].

The decapsulation process ensures that only the intended recipient, who possesses the corresponding private key, can recover the information encrypted within the ciphertext. This procedure is formally defined in **Algorithm 3**.

Algorithm 3. BIKE Decapsulation

Input: Parameter private key $(h_0, h_1, \mu, \sigma) \in \mathcal{H}_w \times \mathcal{M}^2$
Ciphertext $c = (c_0|c_1) \in \mathcal{R} \times \mathcal{M}$
Output: Shared key $K' \in \mathcal{K}$

1: Compute the syndrome $s(x) = c_0(x) \cdot h_0(x) \bmod (x^r - 1)$ and s denotes the binary representation of $s(x)$.
2: Construct the parity-check matrix $H = [H_0|H_1]$, where H_0 and H_1 are circulant matrix repsentation of h_0 and h_1.
3: Recover the error vector $e' = \text{BIKE-Flip}(s, H)$.
4: Recover the candidate message $m' = c_1 \oplus \mathrm{L}(e')$.
5: If the error vector passes verification, i.e., if $e' = \mathrm{H}(m', \mu)$, then set $K' = K(m', c)$; otherwise, fall back to $K' \leftarrow \mathrm{K}(\sigma, c)$.

In the event that message recovery fails, the BIKE scheme replaces the decoded message candidate with the fallback value. This mechanism is designed to prevent information leakage, preserve system integrity, and detect tampering or unauthorized modifications to the ciphertext. By enforcing this safeguard, the system ensures that only valid messages are processed, thereby mitigating the risk of accepting manipulated data. A detailed description of the encapsulation and decapsulation processes of the BIKE can be found in [4].

2.1 QC-MDPC Code

QC-MDPC codes combine the concepts of QC and MDPC codes to enhance efficiency in both encoding and decoding processes, as outlined Definitions 1, 2 and 3.

Definition 1. A binary QC code with block number n_0 and block length r is a linear code whose generator matrix consists of circulant block matrices. A QC code with parameters (n_0, k_0) has index n_0, length $n = n_0 \cdot r$, and dimension $k = k_0 \cdot r$ [5].

Definition 2. An (n, r, w) LDPC or MDPC code is a linear code with codeword length n and block length r, and a parity-check matrix H where each row has a constant weight w [14].

Definition 3. A QC-MDPC code (n_0, k_0, r, w) is a QC code (n_0, k_0) with codeword length $n = n_0 \cdot r$, code dimensio $k = k_0 \cdot r$, order r, and a parity-check matrix H with constant row weight $w = O(\sqrt{n})$ [4].

2.2 Hard Computational Problems of the BIKE Scheme

The security of the BIKE scheme is based on the computational intractability of certain mathematical problems. Table 2 presents the formal formulation of the mathematical problems that underpin the security of the BIKE scheme.

Table 2. Hard Computational Problems in the BIKE Scheme

Problem		Detail
Codeword Finding (CF)	Input:	$H \in \mathbb{F}_2^{(n-k)\times n}$, and $t > 0$.
	Goal:	Find $c \in \mathbb{F}_2^n$ such that $\lvert c\rvert = t$ and $c \cdot H^{\mathrm{T}} = 0$.
QC-Codeword Finding (QCCF)	Input:	$h(x) \in \mathcal{R}_{odd}$ and $w \in 2\mathbb{Z}^+$, $w/2$ odd.
	Goal:	Find $(h_0, h_1) \in \mathcal{H}_w$ such that $h_1(x) + h_0(x) \cdot h(x) = 0$.
Syndrome Decoding (SD)	Input:	$H \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$, and $t > 0$.
	Goal:	Find $(e_0, e_1) \in \mathbb{F}_2^n$ such that $\lvert e\rvert \leq t$ and $e \cdot H^{\mathrm{T}} = s$.
QC-Syndrome Decoding (QCSD)	Input:	$(h, s) \in \mathcal{R}_{odd} \times \mathcal{R}_{p(t)}$, and $t > 0$.
	Goal:	Find $(e_0, e_1) \in \mathcal{E}_t$ such that $e_0(x) + e_1(x) \cdot h(x) = s(x)$.

The CFP and SDP are well-known to be NP-hard in their general form, which means that no polynomial-time algorithms are known to solve them efficiently. In particular, the SD problem has been formally proven to be NP-complete, and current best-known algorithms for solving it, such as Information Set Decoding (ISD) and its variants, still require exponential time in the code length. The quasi-cyclic variants, namely QCCF and QCSD, inherit these hardness assumptions while enabling more compact representations that are suitable for cryptographic applications.

The security of QCCFP and QCSDP is determined not only by their mathematical structure but also by the extent to which an adversary can exploit public information to extract secret information. To evaluate the resilience of these schemes, two primary attack models are employed One-Way Security (OW) and Indistinguishability (IND). The One-Way Security model measures the probability of an adversary A successfully inverting the process, such as reconstructing the secret key (h_0, h_1) or identifying a valid error pair (e_0, e_1), based on the available public information, consisting h and s. If this probability is low, the scheme is considered secure against One-Way attacks. The adversary's advantage in attacking the QCCF scheme is defined as:

$$\mathrm{Adv}_{\mathrm{QCCF}}^{\mathrm{OW}}(A) = \Pr\left[\mathrm{QCCF}(A(h), h) \mid (h_0, h_1) \xleftarrow{\$} \mathcal{H}_w\right].$$

The adversary's advantage in attacking the QCSD scheme is defined as:

$$\mathrm{Adv}_{\mathrm{QCSD}}^{\mathrm{OW}}(A) = \Pr\left[\mathrm{QCSD}(A(h, e_0 + e_1 h), h, e_0 + e_1 h) \mid (h, (e_0, e_1)) \xleftarrow{\$} \mathcal{R}_{odd} \times \mathcal{E}_t\right].$$

Meanwhile, the Indistinguishability model evaluates the extent to which an adversary can distinguish elements produced by the scheme from elements drawn from a random distribution. In other words, the adversary faces the task of classifying whether a given output originates from the scheme's process or from a random distribution. If this distinguishing advantage is low, the scheme is regarded as secure against indistinguishability attacks. The adversary's advantage in attacking the QCCF scheme is defined as:

$$\mathrm{Adv}_{\mathrm{QCCF}}^{\mathrm{IND}}(D) = \left|\Pr\left[D(h_1 h_0^{-1}) \mid (h_0, h_1) \xleftarrow{\$} \mathcal{H}_w\right] - \Pr\left[D(h) \mid h \xleftarrow{\$} \mathcal{R}_{odd}\right]\right|.$$

The adversary's advantage in attacking the QCSD scheme is defined as:

$$\mathrm{Adv}^{\mathrm{IND}}_{\mathrm{QCSD}}(D) = \left|\Pr\left[D(h, e_0 + e_1 h) \mid (h, (e_0, e_1)) \xleftarrow{\$} \mathcal{R}_{odd} \times \mathcal{E}_t\right] - \Pr\left[D(h, s) \mid (h, s) \xleftarrow{\$} \mathcal{R}_{odd} \times \mathcal{R}_{\mathcal{P}(t)}\right]\right|.$$

A detailed description of the computational problems underlying the BIKE can be found in [4] and [8].

2.3 Construction of the BIKE Scheme

The Niederreiter framework is a code-based approach derived from the dual of the McEliece scheme [15]. This approach leverages the linear properties and duality between generator matrices and parity-check matrices, thereby enabling efficient modeling of code-based systems [8]. Meanwhile, FO $^{\not\perp}$ transformation was developed to enhance the security of schemes that satisfy Indistinguishability under Chosen-Plaintext Attack (IND-CPA) into KEM that satisfy IND-CCA. This is achieved through an implicit-rejection mechanism that discards invalid ciphertexts without revealing additional information [9].

3. RESULTS AND DISCUSSION

This study presents a mathematical analysis of the QC-MDPC codes as the foundational basis, as well as the underlying hard computational problems, within the structural construction of the BIKE scheme. The interrelation between the mathematical concepts involved in constructing the BIKE is illustrated in Fig. 2.

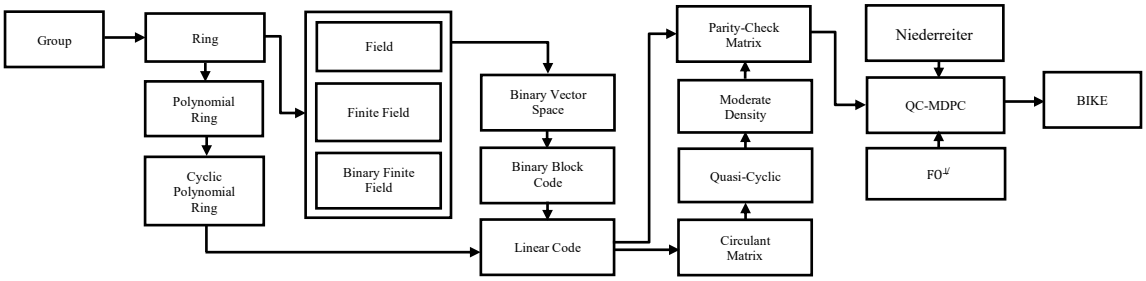

Figure 2. Conceptual Foundation of the BIKE Scheme

A comprehensive explanation of abstract algebra and coding theory can be found in [16] and [17].

3.1 Construction of QC-MDPC Codes

QC-MDPC codes combine the quasi-cyclic property with moderate-density parity-check matrices, forming the mathematical foundation of BIKE, as illustrated in the mathematical concept diagram in Fig. 2. The construction begins with the additive group algebra structure $(\mathbb{F}_2, +)$ of binary numbers, which is extended to the polynomial ring $(\mathbb{F}_2[x], +, \cdot)$ with coefficients in $\mathbb{F}_2$. This structure is further developed into the binary finite field $\mathbb{F}_{2^m}$, serving as the foundation for the binary vector space $\mathbb{F}_2^n$, where all codewords reside. In the space $\mathbb{F}_2^n$, a binary linear block code is defined as a k-dimensional subspace, denoted as a code (n, k), representing the set of codewords $C \subseteq \mathbb{F}_2^n$. Each codeword $c \in C$ is formed as a linear combination of k basis vectors with the generator matrix $G \in \mathbb{F}_2^{k \times n}$ using the relation $c = m \cdot G$ for $m \in \mathbb{F}_2^k$.

The code employed in BIKE possesses a QC structure, as described in Definition 1. In this definition, a binary QC code with index n_0 and order r is a linear code whose generator matrix is composed of circulant block matrices. This implies that the generator matrix is partitioned into n_0 blocks, each of size $k_0 \times r$, where each block is a circulant matrix. The parameters (n_0, k_0) indicate a code length of $n = n_0 \cdot r$ and a dimension of $k = k_0 \cdot r$. The key property of QC codes is that a cyclic shift of a codeword by r positions yield another valid codeword. In polynomial form, if $h_0(x)$ represents the first row of a circulant matrix block, then the cyclic shift $h_0(x) \cdot x^i \bmod (x^r - 1)$ still generates a valid codeword. The parity-check matrix H is constructed from two circulant blocks H_0 and H_1 of size $r \times r$, represented as $H = [H_0 \mid H_1]$, where each circulant block can be fully described by its first row. In BIKE, the process of verifying codewords explicitly avoids the use of a generator matrix and instead relies on the parity-check matrix $H \in \mathbb{F}_2^{(n-k) \times n}$ ensuring code validity via the condition $c \cdot H^{\mathrm{T}} = 0$. This representation significantly reduces storage complexity.

Furthermore, the matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ adheres to moderate density properties, classifying it as an MDPC matrix. As defined in Definition 2, an (n, r, w) LDPC or MDPC code is a linear code of length n, codimension r and a parity-check matrix H with constant row weight w. While LDPC codes typically use a small w with complexity $O(1)$ to facilitate sparse-graph decoding, MDPC codes increase the weight to $O(\sqrt{n})$, which grows with the code length, to enhance resistance to decoding attacks based on solving sparse systems of equations. This representation allows for efficient algorithmic operations like encoding and decoding while providing stronger security against structural attacks compared to traditional LDPC codes.

The integration of QC and MDPC concepts leads to the formulation of QC-MDPC codes with parameters (n_0, k_0, r, w), where the code possesses quasi-cyclic structure, length $n = n_0 \cdot r$, dimension $k = k_0 \cdot r, n_0$, order r, and a parity-check matrix H with constant row weight $w = O(\sqrt{n})$, as previously described in Definition 3. This hybrid approach enables the BIKE scheme to produce a code-based encryption system with compact key sizes, high computational efficiency, and strong resilience against decoding attacks and code structure exploitation.

3.2 Construction Based on Hard Computational Problems

The security of the BIKE scheme is based on two fundamental problems in coding theory: the CF problem and the SD problem. These problems are further developed into QC-specific variants within the quasi-cyclic structure, namely the QCCF problem and the QCSD problem. The relationships among these problems are illustrated in Fig. 3, and their formal definitions are summarized in Table 2.

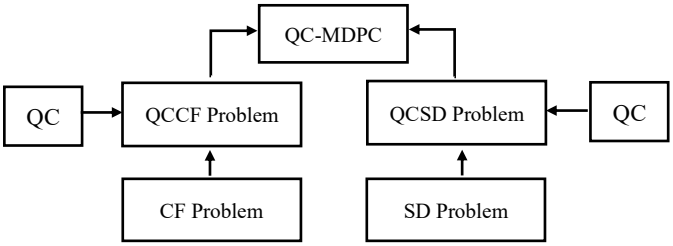

Figure 3. Construction of QC-MDPC Codes Based on Hard Computational Problems

In the QCCF Problem, the main challenge lies in finding the private key pair (h_0, h_1) from the public key $h(x) = h_1(x) \cdot h_0^{-1}(x) \bmod (x^r - 1) \in \mathbb{F}_2[x]/(x^r - 1)$. This problem can be viewed as a generalization of the CF Problem, which involves finding a binary vector $c \in \mathbb{F}_2^n$ of fixed weight $|c| = t$ that satisfies the equation $c \cdot H^{\mathrm{T}} = 0$, where $H \in \mathbb{F}_2^{r\times n}$ is a parity-check matrix. In the QC-MDPC scheme, the QC structure ensures that the private key is composed of two cyclic polynomials $h_0(x)$ and $h_1(x)$, each of degree r and low weight $w/2$, such that the relation $h_1(x) + h_0(x) \cdot h(x) = 0 \bmod (x^r - 1)$ holds.

Meanwhile, the QCSD Problem is an extension of the SD Problem into the domain of cyclic polynomials. In its general form, the SD Problem seeks an error vector $e \in \mathbb{F}_2^n$ of limited weight that satisfies the syndrome equation $s = e \cdot H^{\mathrm{T}}$. Within the context of QC-MDPC, the error is represented as a pair of polynomials (e_0, e_1) that satisfy $e_0(x) + e_1(x) \cdot h(x) \bmod (x^r - 1) = s(x)$, where $h(x)$ is constructed from sparse polynomials $h_0(x)$ and $h_1(x)$ with the relation $h(x) = h_1(x) \cdot h_0^{-1}(x) \bmod (x^r - 1)$. This replaces the role of the matrix H in the QC structure. Consequently, the relationship between SD and QCSD can be expressed as:

$$\begin{aligned} & e_0(x) + e_1(x) \cdot h(x) = s(x), \\ \Longrightarrow\ & e_0(x) + e_1(x) \cdot \left(h_1(x) \cdot h_0^{-1}(x)\right) = s(x), \\ \Longrightarrow\ & e_0(x) \cdot h_0(x) + e_1(x) \cdot h_1(x) = s(x) \cdot h_0(x). \end{aligned} \tag{1}$$

The Eq. (1) shows the QCSD formulation using the polynomial representation. Equivalently, when expressed in matrix-vector form, the relation is given in Eq. (2):

$$e_0 \cdot H_0 + e_1 \cdot H_1 = s \cdot H_0, \tag{2}$$

where e_0 and e_1 are binary vectors corresponding to the error polynomials, and H_0, H_1 are circulant matrices derived from $h_0(x)$ and $h_1(x)$.

This relation is simply the expansion of the vector–matrix multiplication, expressed in block-matrix form as follows:

$$(e_0, e_1)\begin{pmatrix} H_0 \\ H_1 \end{pmatrix} = s \cdot H_0.$$
$$e \cdot H^{\mathrm{T}} = s \cdot H_0.$$

the value $s \cdot H_0$ is computed using $c_0 \cdot H_0$ where $c_0(x) = s(x) = e_0(x) + e_1(x) \cdot h(x)$, such that $s \cdot H_0 = c_0 \cdot H_0 = e \cdot H^{\mathrm{T}}$. To demonstrate the equivalence between Eqs. (1) and (2) namely that $e_0(x) \cdot h_0(x) + e_1(x) \cdot h_1(x) = e_0 \cdot H_0 + e_1 \cdot H_1$, let us assume $h_0(x) = h_{0,0} + h_{0,1}x + \cdots + h_{0,r-1}x^{r-1}$ and $e_0(x) = e_{0,0} + e_{0,1}x + \cdots + e_{0,r-1}x^{r-1}$ both elements of $\mathbb{F}_2[x]/(x^r - 1)$. The circulant matrix H_0, derived from the coefficients of the polynomial h_0 is defined as follows:

$$H_0 = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,r-1} \\ h_{0,r-1} & h_{0,0} & \cdots & h_{0,r-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{0,1} & h_{0,2} & \cdots & h_{0,0} \end{pmatrix},$$

the product $e_0(x) \cdot h_0(x) \bmod x^r - 1$ can be expressed as $e_0(x) \cdot h_0(x) \equiv \sum_{k=0}^{r-1} c_k\, x^k \equiv c_0x^0 + c_1x^1 + c_2x^2 + \cdots + c_{r-1}x^{r-1}$, where the coefficients c_k calculated using Eq. (3):

$$c_k = \sum_{i=0}^{r-1} e_{0,i} \cdot h_{0,(k-i) \bmod r} = e_{0,0}h_{0,(k-0) \bmod r} + e_{0,1}h_{0,(k-1) \bmod r} + \cdots + e_{0,r-1}h_{0,(k-(r-1)) \bmod r}, \quad (3)$$

the multiplication $e_0 \cdot H_0$ yields a vector $c = (c'_0, c'_1, \ldots, c'_{r-1})$, where each c'_k for $k = 0, 1, \ldots, r-1$ is computed using Eq. (4):

$$c'_k = \sum_{i=0}^{r-1} e_{0,i} \cdot (H_0)_{i,k} = e_{0,0}(H_0)_{0,k} + e_{0,1}(H_0)_{1,k} + e_{0,2}(H_0)_{2,k} + \cdots + e_{0,r-1}(H_0)_{r-1,k} \quad (4)$$

due to the circulant property of H_0, every element at position (i, k) satisfies $(H_0)_{i,k} = h_{0,(k-i) \bmod r}$, hence $c'_k = \sum_{i=0}^{r-1} e_{0,i}(H_0)_{i,k} = \sum_{i=0}^{r-1} e_{0,i}\, h_{0,(k-i) \bmod r}$. Therefore, from Eqs. (3) and (4) it is evident that $c'_k = c_k$, which confirms the equality between Eqs. (1) and (2), i.e., $e_0(x) \cdot h_0(x) + e_1(x) \cdot h_1(x) = e_0 \cdot H_0 + e_1 \cdot H_1$.

In the OW-QCCFP scheme, the adversary A is given an instance $h \in \mathcal{H}_w$, where $h(x) = h_1(x)h_0^{-1}(x)$. The adversary then attempts to compute a preimage using only h, i.e., it evaluates $A(h) = A(h_1h_0^{-1})$ by searching for polynomials $h_1'(x)$ and $h_1'(x)$ such that $h(x) = h_1'(x)h_0'^{-1}(x)$. After producing a candidate, the QCCF verification algorithm checks whether the adversary's output is a valid solution, namely whether it satisfies $h_1(x) + h_0(x)h(x) = 0$.

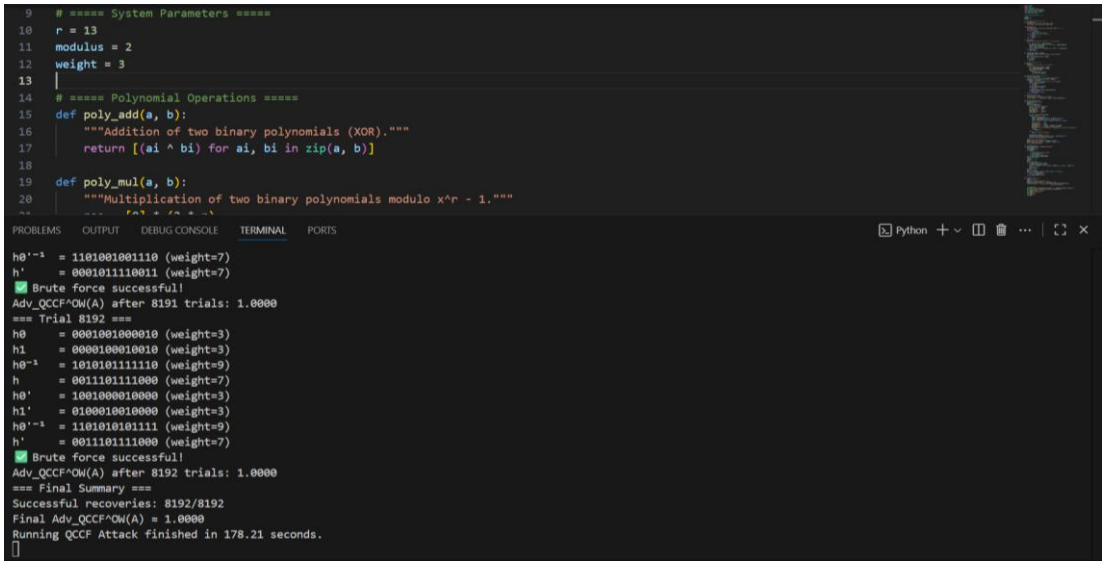

Figure 4. Experimental Evaluation of the OW-QCCF Scheme under Small Parameter Settings

As an illustration, Fig. 4 reports the outcome of OW-QCCF scheme carried out with the small parameter $r = 13$ over 2^{13} trials. The experiment yields $\mathrm{Adv}^{\mathrm{OW}}_{\mathrm{QCCF}}(A) = 1$, indicating that the adversary A can invert the public-key $h = h_1 h_0^{-1}$ and recover (h_0, h_1) by brute force. This is attributable to the small choice of r, which keeps the number of candidate key pairs sufficiently and makes a full search feasible.

For comparison, the simulation in Fig. 4 with $r = 13$ and 2^{13} trials take about 178.21 s to execute. If we adopt the NIST Level-1 parameter $r = 12323$, the number of trials scales to 2^{12323}. Under the same exponential time model, the estimated runtime becomes $T(12323) = 178.21 \times 2^{12310}$. Converting to base-10 using $\log_{10}(2^{12310}) = 12310 \times \log_{10}(2) \approx 12310 \times 0{,}30103 \approx 3706{,}6$. So, estimate is $T(12323) = 178.21 \times 10^{3706.6}$ s. This estimate is astronomically beyond any realistic computational budget (the age of the universe is only 4.35×10^{17}s). Hence, with the standard parameter level 1 $r = 12323$, BIKE is computationally secure against brute force. An analogous observation holds for the OW-QCSDP experiment under small parameters, where the simulation also attains $\mathrm{Adv}^{\mathrm{OW}}_{\mathrm{QCSD}}(A) = 1$.

In the IND-QCCDP scheme, D receives a value h, but the source of this value can originate from two possibilities, namely the value derived from QCCF: $h(x) = h_1(x)h_0^{-1}(x)$, with $(h_0, h_1) \xleftarrow{\$} \mathcal{H}_w$, or a random value drawn from the distribution: $h \xleftarrow{\$} \mathcal{R}_{odd}$. Subsequently, D analyzes the pattern or structure within h and attempts to determine whether the value originates from the computation $h_1 h_0^{-1}$ with $(h_0, h_1) \xleftarrow{\$} \mathcal{H}_w$ or from $h \xleftarrow{\$} \mathcal{R}_{odd}$. After obtaining the result, the IND-QCCF algorithm evaluates how effectively D distinguishes between the two distributions. The simulation results of the IND-QCCF attack using small parameters, specifically $r = 13$, with a total number of 2^{13} trials, are presented in Table 3.

Table 3. Experimental Evaluation of the IND-QCCF Scheme under Small Parameter Settings

Trial	Distribution h	h_0	h_1	h_0^{-1}	h	$D(h)$
1	QCCF	0001000100100	0000100100010	1001110011010	1110111110100	0
2	*Random*	-	-	-	1011011001000	1
3	QCCF	0100000000101	0001100000001	1111001111001	0100011011011	1
4	*Random*	-	-	-	1110011001110	0
5	QCCF	0100000100100	0001100010000	1100111001010	0011111111100	0
6	*Random*	-	-	-	0000100011110	0
7	QCCF	0001000101000	0000001011000	0110011111000	0110010111001	1
8	*Random*	-	-	-	0010101011111	0
9	QCCF	0000001001100	0110100000000	1001101111000	1110001010101	1
10	*Random*	-	-	-	0100110100100	0
...	...	...	...	...	...	...
8192	*Random*	-	-	-	1100010000100	0

Given the input h_0 and h_1 chosen such that each has a Hamming weight of $|h_0| = |h_1| = 3$, the estimated Hamming weight of $h = h_1 h_0^{-1}$ is $r/2 = 13/2$, which lies between the values $|h| = 6$ or $|h| = 7$. This simulation employs a method to check whether the value h generated from the QCCF construction tends to exhibit a specific structural pattern compared to random polynomials, classified according to the criteria:

1. If the Hamming weight of h lies within the range $[\frac{r-1}{2}, \frac{r+1}{2}] = [6,7]$, then D outputs the value 1.
2. If the Hamming weight of h lies outside this range, then D outputs the value 0.

The interpretation of the output value $D(h)$ can be described as follows:

1. Value 1: The adversary D infers that h originates from the QCCF construction.
2. Value 0: The adversary D infers that h is a random element from $\mathcal{R}_{odd}$.

The adversary's advantage in distinguishing between the two distributions is calculated using the formula:

$$\left|\Pr[D(h = h_1 h_0^{-1}) = 1] - \Pr[D(h \sim \mathcal{R}_{odd}) = 1]\right|$$

Hence, based on the experimental results in Table 3, we obtain:

$$\Pr[D(h = h_1 h_0^{-1}) = 1] = 0.22 \text{ and } \Pr[D(h \sim \mathcal{R}_{odd}) = 1] = 0.20$$

$$\text{thus } \mathrm{Adv}^{\mathrm{IND}}_{\mathrm{QCCF}}(D) = |0.22 - 0.20| = 0.02.$$

The value $\mathrm{Adv}^{\mathrm{IND}}_{\mathrm{QCCF}}(D) = 0.02$ indicates that adversary D is unable to effectively distinguish between elements generated from the construction $(h_0, h_1) \xleftarrow{\$} \mathcal{H}_w$ and elements $h \xleftarrow{\$} \mathcal{R}_{odd}$. This limitation arises from the distinguishing assumption employed, which relies solely on the Hamming weight, and is proven insufficient to exploit structural distribution differences in the QCCF scheme. A similar result is observed in the case of IND-QCSD, where adversary D likewise fails to distinguish effectively between elements derived from $\big(h, (e_0, e_1)\big) \xleftarrow{\$} \mathcal{R}_{odd} \times \mathcal{E}_t$ and those sampled from $(h, s) \xleftarrow{\$} \mathcal{R}_{odd} \times \mathcal{R}_{\mathcal{P}(t)}$.

The QCCF and QCSD problems, as presented in Table 2 and the preceding evaluation, constitute the primary security assumptions of the BIKE scheme. To date, no classical or quantum algorithms have been able to solve these problems efficiently. Consequently, the security of BIKE relies on the computational hardness of QCCF and QCSD, which are implemented through the Niederreiter framework and the $\mathrm{FO}^{\not\perp}$ transformation.

3.3 Construction Based on the BIKE Scheme Framework

The BIKE scheme is constructed based on the Niederreiter framework and the $\mathrm{FO}^{\not\perp}$ transformation, as illustrated in Fig. 2. These two constructions form the foundational structure for the parameter setup, key generation, encapsulation, and decapsulation processes employed in the BIKE, as specified in Table 1, **Algorithms 1**, **2** and **3**.

The Niederreiter framework is an approach that utilizes the dual code of the McEliece cryptosystem [15]. In the McEliece scheme, a message m is encrypted by multiplying it with a generator matrix G and adding an error vector e, resulting in the ciphertext $c = m \cdot G + e$ [8], [14]. In contrast, the Niederreiter scheme performs encryption by multiplying a random binary vector x with a parity-check matrix H, producing the ciphertext $c = H \cdot x^{\mathrm{T}}$ [8]. Both schemes are related through the use of linear codes satisfying the orthogonality condition $G \cdot H^{\mathrm{T}} = 0$, establishing Niederreiter as the dual of McEliece [17]. This duality provides a foundation for BIKE's key and ciphertext construction, which is then adapted to enhance efficiency and security.

Within the Niederreiter framework, BIKE v5.2 uses sparse polynomials for private keys, compresses the public key into a single polynomial, and constructs ciphertexts from a syndrome combined with an obfuscated hash. Decoding is performed using the BIKE-Flip algorithm, which refines bit-flipping with a dynamic threshold to improve accuracy and reduce predictable failure patterns. The rationale behind the BIKE parameter design based on The Niederreiter framework is grounded in several critical considerations. which are further explained below.

The private key consists of two sparse polynomials, $(h_0(x), h_1(x))$ generated using the FY-CWW algorithm with a uniform variant. A detailed explanation of the FY-CWW algorithm can be found in [4]. The uniform distribution ensures that all possible keys have an equal probability of being selected. In this context, variable execution time is acceptable, as key generation is performed only once and does not influence the runtime behavior observable by an attacker. The polynomials h_0 and h_1 must possess fixed weight and random distribution. When h_0 and h_1 have fixed weights and are randomly distributed, the resulting public key h and the codeword c will also be randomly distributed. Consequently, the derived syndrome s follows a uniform distribution, making it difficult for adversaries to predict. In contrast, if h_0 and h_1 lack fixed weights or exhibit non-random distributions, the resulting h will exhibit certain patterns, leading to patterned c, which in turn results in a predictable s. Such predictability opens up the possibility for exploitation attacks on the underlying structure. Each polynomial has a weight of $w/2$, so their combined weight satisfies $|h_0| + |h_1| = w$. Equal weighting between h_0 and h_1 is crucial for maintaining a balanced syndrome distribution during the decoding process. In this system, the parity-check matrix is defined as $H = [H_0 \mid H_1]$. When a codeword vector c is multiplied with H, it produces $s = c \cdot H$. For the bits in s to be uniformly distributed, the weights of h_0 and h_1 must be equal. Any imbalance in weight may lead to a non-uniform syndrome distribution, potentially degrading the performance of the decoding algorithm and increasing the likelihood of errors in code correction. Furthermore, $h_0(x)$ and $h_1(x)$ must have a maximum degree less than r, where $r \in P$ is a prime number satisfying $\mathrm{ord}_r(2) = r - 1$, ensuring that 2 is a primitive element modulo r. This design prevents non-trivial factors in $\mathbb{F}_2$. If r is prime, $x^r - 1$ has only trivial factors, reducing the risk of algebraic attacks. Additionally, the sparse structure of h_0 and h_1 ensures a low density, minimizing the number of required operations and helping maintain a low DFR.

The public key $h(x)$ in BIKE is defined as $h(x) = h_1(x) \cdot h_0^{-1}(x)$. Storing only a single polynomial $h(x)$ instead of the pair (h_0, h_1) reduces the public key size, optimizing storage and transmission. Since $h_0(x)$ is a binary cyclic polynomial over $\mathbb{F}_2$, computing $h_0^{-1}(x)$ without knowing $h_0(x)$ is computationally infeasible, making it difficult for an attacker to recover $h_0(x)$ or $h_1(x)$ from $h(x)$ independently.

The ciphertext in BIKE consists of two components: $c_0(x) = e_0(x) + e_1(x) \cdot h(x)$ and $c_1 = m \oplus L(e_0, e_1)$. The computation of $e_0(x) + e_1(x) \cdot h(x)$ forms a syndrome as described in Table 2, directly used as the ciphertext component c_0, which the BIKE-Flip decoder processes without reconstructing the error vector. This contrasts with the McEliece scheme, where the ciphertext $c = m \cdot G + z$ explicitly stores the error vector z, increasing the ciphertext size by the codeword length n. McEliece decryption also involves computing the permutation matrix inverse P^{-1}, multiplying it with c, and then passing it to the decoder, adding complexity. In contrast, BIKE only requires multiplication with h_0 before decoding, avoiding matrix inversion and large matrix multiplications. By storing only the syndrome in the ciphertext, BIKE achieves smaller ciphertext size and faster decapsulation compared to McEliece. The second ciphertext component, c_1, is obtained by XORing the plaintext m with the hash of $L(e_0, e_1)$, adding an additional layer of masking to m. This prevents adversaries from directly extracting m from the ciphertext without recovering e_0 and e_1, enhancing message security.

The decoder in BIKE, known as BIKE-Flip, employs a bit-flipping method that is more efficient for QC-MDPC codes than list-decoding or information set decoding (ISD). Bit-flipping exploits the sparsity of the code structure, whereas ISD relies on brute-force guessing of error-free bit subsets, which scales exponentially with the number of errors t as 2^t. List-decoding, which seeks all codewords within a certain distance, is also inefficient for sparse codes. Since QC-MDPC codes have a sparse parity-check matrix with only $w = \sqrt{n}$ elements set to one, each bit-flipping iteration requires roughly $O(\sqrt{n})$ time, making it a faster and more practical approach for large QC-MDPC codes.

This bit-flipping strategy is further refined in BIKE-Flip, the decoder BIKE-Flip is used in BIKE v5.2, which is superior to the decoder used in BIKEv5.1, namely Black-Gray-Flip (BGF). BIKE-Flip has advantages over BGF in terms of structure and operational mechanism. BIKE-Flip uses 7 iterations, whereas BGF uses 5 iterations. BGF utilizes the BFMaskedIter and BFIter procedures with masking, while BIKE-Flip only employs BFIter without masking. In addition, BGF uses a fixed threshold, whereas BIKE-Flip applies a dynamic threshold. Computationally, BGF is more complex due to its two procedures and masking, while BIKE-Flip is simpler. As shown in Fig. 4, experimental results indicate that BIKE-Flip achieves faster execution time despite requiring more iterations. This improvement is attributed to the elimination of the BFMaskedIter procedure and the masking mechanism in BIKE-Flip, thereby relying only on BFIter, which has proven to be more efficient in terms of execution time. Furthermore, as illustrated in Fig. 5, BIKE-Flip demonstrates lower memory consumption compared to BGF. This is because BIKE-Flip does not require the masking mechanism nor the storage of masking values. Instead, BIKE-Flip leverages a more efficient threshold computation.

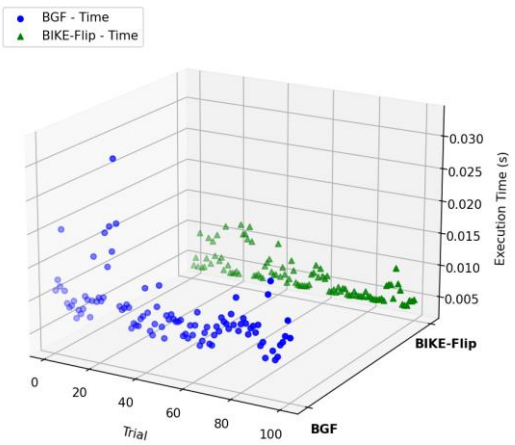

Figure 5. Execution Time Comparison of BIKE-Flip and BGF

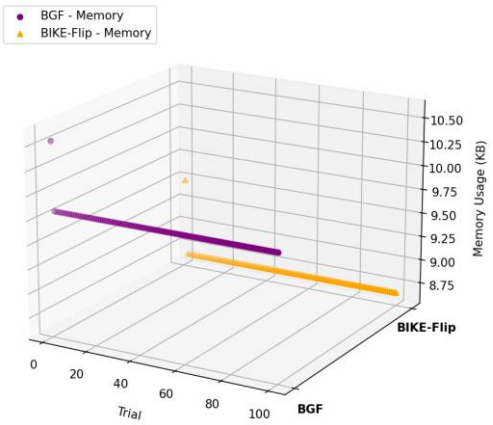

Figure 6. Memory Usage Comparison of BIKE-Flip and BGF

A dynamic threshold is used to improve decoding success. BGF applies a constant threshold at each iteration, which may lead to errors due to propagation from previous iterations, making decoding failure patterns more predictable. As a solution, BIKE-Flip applies a dynamic threshold dependent on the initial syndrome weight S_0, the current syndrome weight S, and the iteration i. The dynamic threshold is updated in each iteration to reduce the probability of unnecessary flipping.

In the first iteration, the threshold is defined as $T_1(S_0) = f_t(S_0) + \delta$, where the threshold is set higher to allow for the correction of more errors. In the second iteration, the threshold becomes $T_2(S_0) = \frac{1}{3}(2 \cdot f_t(S_0) + \frac{d+1}{2}) + \delta$, decreasing gradually to balance bit flipping and decoding stability. In the third iteration, the threshold is expressed as $T_3(S_0) = \frac{1}{3}(f_t(S_0) + 2 \cdot \frac{d+1}{2}) + \delta$, approaching the minimum value to adjust to the remaining syndrome. For iterations $i \geq 4$, the threshold stabilizes at $T_i(S_0) = \frac{d+1}{2} + \delta$, corresponding to the average syndrome weight since most errors have been corrected.

A higher threshold in the first iteration allows for the correction of more errors. As the threshold value increases in the following iterations, the decoding failure pattern becomes smaller, reducing the possibility of exploitation by an attacker. Further explanation of BIKE-Flip can be found in [4], and the previous decoder is discussed in [15].

The application of the Niederreiter framework in BIKE v5.2 enables the construction of a mathematically structured and efficient code-based cryptographic system. The private key consists of a pair of sparse polynomials with fixed weight and random distribution, ensuring a well-balanced and unpredictable syndrome distribution. The public key is compressed into a single polynomial via inversion, reducing storage requirements while concealing the internal structure. The ciphertext comprises a syndrome resulting from a linear combination of the error vector and the public key, along with an obfuscated hash value, effectively protecting the message from direct ciphertext attacks. The BIKE-Flip decoder employed in this version adopts a dynamic threshold to control the bit-flipping process at each iteration, leading to improved error correction accuracy and reducing the risk of error-pattern exploitation. Through a combination of secure key design, compact ciphertext, and adaptive decoding, BIKE v5.2 demonstrates reliable efficiency, making it a relevant candidate for post-quantum, code-based cryptographic solutions.

Despite leveraging QC-MDPC codes, BIKE exhibits limitations in resisting IND-CCA attacks. The Niederreiter framework lacks intrinsic mechanisms to protect the ciphertext from tampering by adversaries [18]. In such attacks, an adversary may alter the ciphertext and glean information about the private key. To address this vulnerability, BIKE incorporates the $\text{FO}^{\not\perp}$ transformation, which serves to strengthen security and provide additional protection.

The $\text{FO}^{\not\perp}$ transformation, developed by Fujisaki and Okamoto, is designed to enhance the security of public-key encryption schemes by converting an IND-CPA-secure encryption scheme into a KEM resistant to IND-CCA attacks [9]. With $\text{FO}^{\not\perp}$, even if an adversary is able to submit modified ciphertexts and observe the decryption outputs, they are unable to derive any information about the private key or the plaintext. The integration of $\text{FO}^{\not\perp}$ into BIKE ensures that the scheme is not only efficient in terms of key generation, encapsulation, and decapsulation but also more resilient against IND-CCA attacks, reaction attacks, and multi-target key attacks [4].

This enhancement is further reinforced in the BIKE scheme with FO$^{\not\perp}$, the error vector (e_0, e_1) is computed using the formula $(e_0, e_1) = H(m, \mu)$, where $\mu = \pi_l(h)$ is the portion of the public key h used for binding. Specifically, the function $\pi_l(h)$ refers to taking the first 256 bits of the public key h. Without binding to h, such that $(e_0, e_1) = H(m)$, the hash function output depends solely on m. In this case, an attacker could precompute a list of (e_0, e_1) for various values of m, making a multi-target key attack possible, as the same set of (e_0, e_1) could be used for different values of h. A detailed explanation of the multi-target key attack is available in [19]. Conversely, when binding to h by computing $(e_0, e_1) = H(m, \mu)$, the hash result becomes unique for each h, as $\pi_l(h) \neq \pi_l(h')$ when $h \neq h'$. As a result, attackers cannot reuse the same set of (e_0, e_1) across different values of h. The error vector (e_0, e_1) is generated using the FY-CWW algorithm with a fixed-time variant to eliminate the possibility of timing attacks. In contrast, the private key polynomials (h_0, h_1) are generated using a uniform variant. A detailed explanation of timing attacks can be found in [20].

The output of the BIKE scheme is a shared key that will be used as the secret key between the sender and the receiver. This shared key K plays a crucial role in ensuring the security of the key exchange process. The value of K is determined based on the result of the decryption process of the received ciphertext c. If decoding succeeds, KKK is derived from m′m'm′ and c ; if decoding fails, K is derived from the fallback value σ and c. The rule is defined as follows:

$$K = \begin{cases} \mathrm{K}(m', c), & \text{if decoding succeeds,} \\ \mathrm{K}(\sigma, c), & \text{if decoding fails.} \end{cases}$$

where σ is a fallback value, random and independent of the ciphertext, ensuring the shared key output is always produced and attackers cannot distinguish whether the shared key originates from a valid decoded m' or from the σ. Without a fallback mechanism, the hash function output K could be observed by attackers, potentially revealing information about the private key and enabling reaction attacks. Explanation of reaction attacks can be found in [20].

The hash function H in BIKE uses FY-CWW with SHAKE256 to produce constant-weight vectors. SHAKE256 is based on Keccak, which has been verified by NIST as a secure hash function [21]. SHAKE256 offers the advantage of producing variable-length outputs as needed, avoiding the repeated hashing required in SHA-256 or SHA-384. FY-CWW ensures that exactly w bits in the vector are always set to "1", which is necessary to maintain a fixed distribution and prevent decoding errors during error recovery. Without FY-CWW, the distribution of "1" bits in the vector becomes unpredictable, leading to more frequent decoding failures.

In addition, the hash function K uses SHA-384 to generate the shared key in communication sessions. The explanation is as follows: SHA-384 has been verified by NIST and meets the criteria for secure hash functions, ensuring that the shared key cannot be predicted and attackers cannot reconstruct the input from the hash output. Explanation of SHA-384 can be found in [21]. SHA-384 is chosen because it offers a balance between security and efficiency in modern cryptographic schemes. With a 384-bit output size, it provides a higher security level than SHA-256 but with less overhead than SHA-512. The hash function K is used to generate the final shared key to be used in communication. If the pre-image resistance of K is weak, attackers could attempt to reverse the hashing process to obtain the plaintext m or ciphertext c, which could lead to attacks on the KEM scheme.

Complementing this, the hash function L also uses SHA-384 but serves a different purpose than K. While K is used to generate the final shared key in a communication session, L functions in the key derivation process to produce intermediate values that support the overall security of the scheme. The key difference between L and K lies in the type of input processed and their respective roles in the scheme. The hash function L does not directly produce the shared key, but rather ensures the integrity and security of intermediate values during key derivation. Therefore, besides pre-image resistance, L must also have strong second pre-image resistance to prevent other inputs from producing the same hash. If second pre-image resistance is weak, there is a risk that intermediate values can be forged or predicted, potentially compromising the security of the scheme. With its security properties verified by NIST, SHA-384 ensures that K and L are strongly protected against cryptographic attacks, making it a suitable choice for the BIKE scheme. Explanation of the security properties of hash functions: pre-image resistance, second pre-image resistance, and collision resistance, can be found in [1].

The BIKE scheme is designed based on the Niederreiter construction and the FO$^{\not\perp}$ transformation to establish an efficient and attack-resistant KEM. By leveraging QC-MDPC codes within the Niederreiter

framework characterized by sparse structure and fixed weight BIKE enables the efficient generation of public and private keys, reduces ciphertext size, and accelerates decoding through the BIKE-Flip algorithm with a dynamic threshold. The $FO^{\not\perp}$ transformation enhances security against IND-CCA, reaction attacks, and multi-target key attacks through the use of binding and fallback mechanisms, as well as the integration of secure hash algorithms such as SHAKE256 and SHA-384.

4. CONCLUSION

This study demonstrates that BIKE v5.2, constructed through the integration of QC-MDPC code structures, the Niederreiter framework, and the $FO^{\not\perp}$ transformation, successfully forms an efficient, secure, and relevant KEM for post-quantum cryptography. Through mathematical analysis, this research confirms its strong security foundation and resilience against both classical and quantum attacks. The findings of this study can be summarized as follows:

1. The BIKE-Flip decoder with dynamic threshold enhances stability, reduces predictable failure patterns, and improves efficiency.
2. NIST-recommended parameters make brute-force attacks computationally infeasible, confirming BIKE's mathematical validity.
3. Efficient key and ciphertext design, along with fallback and secure hash functions, mitigates multi-target and reaction attacks.
4. Despite DFR instability, BIKE remains efficient and secure; this risk can be managed through parameter tuning and fallback mechanisms.

Although BIKE was not selected for NIST standardization, it continues to stand as a strong and relevant candidate for post-quantum cryptography. Future research should focus on improving decoder stability, refining parameter strategies, and exploring integration into hybrid PQC frameworks to ensure practical deployment.

Author Contributions

Rosa Rosa: Conceptualization, Formal Analysis, Writing – Original Draft, Visualization. Sa'aadah Sajjana Carita: Formal Analysis, Methodology, Writing – Review and Editing. Nadia Paramita Retno Adiati: Methodology, Project Administration, Writing – Review and Editing. Sri Rosdiana: Funding Acuisition, Supervision, Writing – Review and Editing. All authors discussed the results together and contributed to the final version of the manuscript.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgment

The authors would like to express their sincere gratitude to the supervising lecturer for the valuable guidance, direction, and constructive feedback provided throughout the research and the preparation of this manuscript. Appreciation is also extended to the Politeknik Siber dan Sandi Negara for providing research facilities and institutional support.

Declarations

The authors declares that he/she has no conflicts of interest to report study.

Declaration of Generative AI and AI-assisted Technologies

Generative AI tools were used solely for language refinement (grammar, spelling, and clarity). The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. The authors reviewed and approved all final text.

REFERENCES

[1] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, HANDBOOK OF APPLIED CRYPTOGRAPHY. CRC Press, 2018.doi: https://doi.org/10.1201/9780429466335

[2] D. M. Pradana, "KAJIAN MATEMATIS MEKANISME ENKAPSULASI KUNCI KYBER.CCAKEM BERBASIS MASALAH LEARNING WITH ERROR," 2023.

[3] G. M. Raimondo and L. E. Locascio, MODULE-LATTICE-BASED KEY-ENCAPSULATION MECHANISM STANDARD. Gaithersburg,: National Institute of Standards and Technology, 2024. doi: https://doi.org/10.6028/NIST.FIPS.203.

[4] N. Aragon., "BIKE: BIT FLIPPING KEY ENCAPSULATION," Internet Draft, 2024. [Online]. Available: https://bikesuite.org/docs/BIKE.pdf [Accessed: Oct. 22, 2024].

[5] M. R. Nosouhi, S. W. A. Shah, L. Pan, and R. Doss, "BIT FLIPPING KEY ENCAPSULATION FOR THE POST-QUANTUM ERA," *IEEE Access*, vol. 11, pp. 56181–56195, 2023. doi: https://doi.org/10.1109/ACCESS.2023.3282928.

[6] National Institute of Standards and Technology, "POST-QUANTUM CRYPTOGRAPHY PROJECT," 2024. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals [Accessed: Oct. 26, 2024].

[7] N. Aragon et al., "OFFICIAL WEB PAGE OF BIKE SUITE," 2024. [Online]. Available: https://bikesuite.org/ [Accessed: Dec. 26, 2024].

[8] V. Vasseur, "POST-QUANTUM CRYPTOGRAPHY: A STUDY OF THE DECODING OF QC-MDPC CODES," Ph.D. dissertation, Université Paris Cité, Paris, France, 2021. [Online]. Available: https://theses.hal.science/tel-04523204v1

[9] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A MODULAR ANALYSIS OF THE FUJISAKI-OKAMOTO TRANSFORMATION," in *Proc. Theory of Cryptography Conf. (TCC)*, Cham, Switzerland: Springer, pp. 341–371, Nov. 2017. Doi: https://doi.org/10.1007/978-3-319-70500-2_12

[10] S. W. A. Shah, M. R. Nosouhi, L. Pan, and R. Doss, "SoK: ON EFFICACY OF THE BGF DECODER FOR QC-MDPC-BASED QUANTUM-SAFE CRYPTOSYSTEMS," in *Proc. 10th ACM Asia Public-Key Cryptography Workshop (AsiaPKC)*, Melbourne, Australia, Jul. 2023, pp. 2–9. doi: https://doi.org/10.1145/3591866.3593070.

[11] G. Alagic et al., "STATUS REPORT ON THE FOURTH ROUND OF THE NIST POST-QUANTUM CRYPTOGRAPHY STANDARDIZATION PROCESS," NIST Internal Report (NIST IR) 8545, Gaithersburg, MD, USA, Mar. 2025. doi: https://doi.org/10.6028/NIST.IR.8545.

[12] M. R. Nosouhi, .yed W. Shah, L. Pan, Y. Zolotavkin, A. Nanda, P. Gauravaram and R. Doss., "WEAK-KEY ANALYSIS FOR BIKE POST-QUANTUM KEY ENCAPSULATION MECHANISM," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 2160–2174, Apr. 2023. [Online]. Available: http://arxiv.org/abs/2204.13885

[13] N. Drucker, S. Gueron, and D. Kostic, "BINDING BIKE ERRORS TO A KEY PAIR," in *Proc. Cryptographers' Track at RSA Conf. (CT-RSA)*, Cham, Switzerland: Springer, pp. 275–281, Jul. 2021.doi: https://doi.org/10.1007/978-3-030-78086-9_21

[14] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-MCELIECE: NEW MCELIECE VARIANTS FROM MODERATE DENSITY PARITY-CHECK CODES," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 2069–2073.doi: https://doi.org/10.1109/ISIT.2013.6620590

[15] M. Mahajan, B. Singh, A. Agrawal, and A. K. Mishral, "COMPARATIVE ANALYSIS OF BIT FLIPPING DECODERS IN BIKE PQC," in *Lecture Notes in Networks and Systems*, vol. 941. Cham, Switzerland: Springer, 2024, pp. 345–356. doi: https://doi.org/10.1007/978-981-99-9531-8_28.

[16] T. W. Judson, ABSTRACT ALGEBRA: THEORY AND APPLICATIONS. Nacogdoches, TX: Stephen F. Austin State Univ., 2020. [Online]. Available: https://scholarworks.sfasu.edu/ebooks

[17] W. C. Huffman and V. Pless, *FUNDAMENTALS OF ERROR-CORRECTING CODES*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[18] I. Von Maurich, L. Heberle, and T. Güneysu, "IND-CCA SECURE HYBRID ENCRYPTION FROM QC-MDPC NIEDERREITER," in *Proc. 7th Int. Workshop on Post-Quantum Cryptography (PQCrypto)*, Fukuoka, Japan, Feb. 2016. [Online]. Available: https://www.nsa.gov/ia/programs/suiteb doi: https://doi.org/10.1007/978-3-319-29360-8_1

[19] T. Wang, A. Wang, and X. Wang, "EXPLORING DECRYPTION FAILURES OF BIKE: NEW CLASS OF WEAK KEYS AND KEY RECOVERY ATTACKS," in *Proc. Int. Cryptology Conf. (CRYPTO)*, Santa Barbara, CA, USA, Aug. 2023. Doi: https://doi.org/10.1007/978-3-031-38548-3_3

[20] Q. Guo, T. Johansson, and P. Stankovski, "A KEY RECOVERY ATTACK ON MDPC WITH CCA SECURITY USING DECODING ERRORS," in *Advances in Cryptology – ASIACRYPT*, Hanoi, Vietnam, Dec. 2016, pp. 789–815.doi: https://doi.org/10.1007/978-3-662-53887-6_29

[21] M. J. Dworkin, "SHA-3 STANDARD: PERMUTATION-BASED HASH AND EXTENDABLE-OUTPUT FUNCTIONS," FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION (FIPS) 202, NIST, Gaithersburg, MD, USA, 2015. doi: https://doi.org/10.6028/NIST.FIPS.202.