IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024TAP0003

Publicized:2024/09/04

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER *Special Section on Information Theory and Its Applications*

Efficient Reconstruction in Key Recovery Attack on the QC-MDPC McEliece Cryptosystems

Motonari OHTSUKA[†], **Takahiro ISHIMARU**[†], **Yuta TSUKIE**[††], ***Nonmembers***,
Shingo KUKITA[†a)], ***and*** **Kohtaro WATANABE**[†b)], ***Members***

SUMMARY Realization of large quantum computers is believed to jeopardize the security of cryptosystems relying on computational complexity of some mathematical problems, such as prime factorization and discrete logarithm problem. In this light, post-quantum cryptography, which is secure even after large quantum computers are realized, has been getting a lot of attention. National Institute of Standards and Technology (NIST) recently started a standardization process for post-quantum cryptosystems. The McEliece public-key cryptosystem based on quasi-cyclic moderate-density parity-check (QC-MDPC) codes is a promising candidate in this NIST standardization. Recently, attacks on the QC-MDPC McEliece scheme have extensively been investigated. The one proposed by Guo et al. exploits statistical information of decoding errors to reconstruct the secret key. This attack is twofold: (1) obtaining the *distance spectrum* of the secret key from statistical information of decoding errors, and (2) reconstructing the secret key from the distance spectrum. The bit-flipping decoding, which is commonly used to decode the QC-MDPC scheme, is considered to be vulnerable to the first part of this attack. Meanwhile the second part of the attack in the original version by Guo et al. requires considerable time because they use recursive search in this part. In this paper, we propose another method to reconstruct the secret key from the obtained distance spectrum on the basis of a method proposed by Fabšič et al. They found that the key construction can be mapped to a clique problem in graph theory. Using their observation, we apply a breadth-first search algorithm to the key reconstruction. Numerical experiments show that our method reconstructs the secret key more efficiently than recursive search in the original key reconstruction proposed by Guo et al.

key words: *QC-MDPC, decoding error rate, distance spectrum, key recovery attack, clique problem*

1. Introduction

The evolution and spread of networks have made cryptography indispensable in modern society. Although cryptosystems based on factoring or discrete logarithm are widely utilized, they can be cracked by a large quantum computer, which can solve both problems in polynomial time [1]. Accordingly, providing new secure cryptosystems in a "post-quantum" world, where large quantum computers are available, are gaining a lot of attention. For example, National Institute of Standards and technology (NIST) initiated a standardization process by calling for proposals of post-quantum cryptosystems [2].

The McEliece public-key cryptosystem is based on the difficulty of decoding a random linear code [3], and has thoroughly been investigated [4]–[6]. The original version with Goppa codes is considered to be still secure; however, its public key size is undesirably large because this scheme uses the whole generator matrix of a linear code as the public key. This motivates us to construct variants with smaller key sizes. A variant based on quasi-cyclic moderate-density parity-check (QC-MDPC) codes was proposed [7] and has been attracting attention. Thanks to the quasi-cyclicity of a generator matrix in this scheme, one can represent it by its first row, and thus the key size can be much smaller than the original version. "Moderate-density" means that a row of the parity-check matrix has more ones than quasi-cyclic low-density parity-check (QC-LDPC) codes, but much less than the length of the row. The QC-MDPC scheme is recognized as a candidate of the NIST standardization.

Recently, an attack on the QC-MDPC scheme was proposed in [8], aiming to reconstruct the secret key of the QC-MDPC scheme from statistical information on decoding errors. This attack is comprised of two parts. The first part aims to acquire the *distance spectrum*, which is the set of distances between any two ones in the secret key. Sending certain messages to a legitimate receiver, and observing receiver's reactions, an adversary can statistically obtain the distance spectrum. The effectiveness of this part depends on the decoding algorithm, and it has been demonstrated that the bit-flipping (BF) decoding [7], [9]–[14], which is commonly used to decode the QC-MDPC scheme, is vulnerable to this [15]: If the legitimate receiver uses the BF decoding, the attacker can easily obtain the distance spectrum. The second part is to reconstruct the secret key from a given distance spectrum. In the original proposal in [8], recursive search was utilized in this part, and thus the key reconstruction was time-consuming in general.

In this paper, we propose another method to accomplish the second part with shorter reconstruction time than the recursive search. Our method is based on the idea by Fabšič et al. [16], which interprets the key reconstruction to a clique problem in graph theory. They showed correspondence between a graph and a distance spectrum, and found that the secret key to be found is represented by a clique (complete subgraph) in the graph. They applied this observation to reconstruct the secret key in the QC-LDPC scheme, in which searching for small cliques is sufficient. We modify their method, and employ it to decode the QC-MDPC scheme. Numerical experiments show that our method significantly reduces the key reconstruction time compared with the original recursive search proposed by Guo et al. We note that

[†]The authors are with Dept. of Computer Science, National Defense Academy of Japan, Kanagawa, 239-8686 Japan
[††]The author is with Japan Air Self-Defense Force
a) E-mail: kukita@nda.ac.jp
b) E-mail: wata@nda.ac.jp

Paiva and Terada proposed another algorithm for the second part of the key recovery [17]; however, this requires an additional information on the secret key compared with ours.

The remaining of this paper is organized as follows. In Sec. 2, we briefly review the QC-MDPC McEliece cryptosystem. The key recovery attack proposed by Guo et al. is explained in Sec. 3. In Sec. 4, we propose the new method for the key reconstruction. Section 5 exhibits numerical results while comparing our method to the original key recovery algorithm with the recursive search. Section 6 is devoted to conclusions.

2. QC-MDPC McEliece cryptosystems

We briefly review the QC-MDPC scheme [7]. This scheme is characterized by a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{r\times n}$ and the corresponding generator matrix $\boldsymbol{G} \in \mathbb{F}_2^{k\times n}$, where k is the information bit length, n the code length, and r the codimension: $r = n-k$. Let t denote the number of correctable errors. The secret key in the QC-MDPC scheme is the parity-check matrix $\boldsymbol{H}$ with the form,

$$\boldsymbol{H} = \left[\boldsymbol{H_0}\ \boldsymbol{H_1}\ \cdots \boldsymbol{H_{n_0-1}}\right]. \tag{1}$$

Here, $\boldsymbol{H_i}$ $(0 \le i \le n_0 - 1)$ is a circulant matrix of size $r \times r$ and hence $n = n_0 r$, $k = (n_0 - 1)r$. We assume that each $\boldsymbol{H_i}$ has the row weight (= the number of ones in a row) d_v, and thus the row weight w of the whole parity-check matrix $\boldsymbol{H}$ is $w = d_v \times n_0$. In the QC-MDPC scheme, $\boldsymbol{H}$ is taken to be sparse and the weight w scales in $O(\sqrt{n \log n})$.

The public key is the generator matrix, which is expressed via $\boldsymbol{H}$ as

$$\boldsymbol{G} = \left[\begin{array}{c|c} & \left(\boldsymbol{H_{n_0-1}^{-1}} \cdot \boldsymbol{H_0}\right)^T \\ \boldsymbol{I_k} & \vdots \\ & \left(\boldsymbol{H_{n_0-1}^{-1}} \cdot \boldsymbol{H_{n_0-2}}\right)^T \end{array}\right]. \tag{2}$$

Thanks to the cyclicity of $\left(\boldsymbol{H_{n_0-1}^{-1}} \cdot \boldsymbol{H_i}\right)^T (0 \le i \le n_0 - 2)$, it is sufficient to publish only the first row of these matrices. Therefore, the size of the public key is considerably small comparing with the original version of McEliece cryptosystems using Goppa codes, whose public key is the whole generator matrix.

A sender encrypts a message $\boldsymbol{m}$ as

$$\boldsymbol{c} = \boldsymbol{mG} + \boldsymbol{e}, \tag{3}$$

where $\boldsymbol{e}$ is a randomly generated error vector with length n and the Hamming weight less than t: $w(\boldsymbol{e}) \le t$. A receiver decrypts the ciphertext $\boldsymbol{c}$ through the following steps.

(i) Operate $\boldsymbol{H}$ on the ciphertext $\boldsymbol{c}$ and obtain the syndrome $\boldsymbol{s}$:

$$\boldsymbol{s} := \boldsymbol{cH}^T = \boldsymbol{mGH}^T + \boldsymbol{eH}^T = \boldsymbol{eH}^T.$$

(ii) Find $\boldsymbol{e}$ from given $\boldsymbol{H}$ and $\boldsymbol{s}$. Due to the sparsity of $\boldsymbol{H}$, one can efficiently solve this problem. Then, the receiver finds $\boldsymbol{e}$ and obtains $\boldsymbol{c'} = \boldsymbol{mG}$.

(iii) Operate $\boldsymbol{G^{-1}}$ on $\boldsymbol{c'}$, and the receiver obtains $\boldsymbol{m}$.

Let us explain why the decryption is difficult for an eavesdropper. As $\boldsymbol{H}$ is secret, the eavesdropper tries to decrypt the message using another parity-check matrix $\widetilde{\boldsymbol{H}}$:

$$\widetilde{\boldsymbol{H}} = \left[(\boldsymbol{H_{n_0-1}^{-1}} \cdot \boldsymbol{H_0}) \cdots (\boldsymbol{H_{n_0-1}^{-1}} \cdot \boldsymbol{H_{n_0-2}}) \mid \boldsymbol{I_k}\right], \tag{4}$$

which can be generated on the basis of the public key $\boldsymbol{G}$. The step (i) is performed similarly to that for the legitimate receiver, and the eavesdropper obtains $\tilde{\boldsymbol{s}} = \boldsymbol{e}\widetilde{\boldsymbol{H}}^T$. In the step (ii), the eavesdropper must find $\boldsymbol{e}$ from $\tilde{\boldsymbol{s}}$ and $\widetilde{\boldsymbol{H}}^T$; however, this is difficult to solve because $\widetilde{\boldsymbol{H}}^T$ is not sparse in general. This problem, called the syndrome decoding problem, is known to be NP-complete (more precisely, NP-equivalent), which provides the foundation for the security of the QC-MDPC scheme [18], [19].

The security of the QC-MDPC scheme is governed by the parameters (n, r, d_v). Table 1 presents the parameters at several security levels proposed in [7]. Also, the parameters for Bit Flipping Key Encapsulation (BIKE) corresponding to the 128-bit security level are shown in Table 2 [20]. BIKE is a key encapsulation mechanism based on QC-MDPC codes. This scheme has been submitted to the NIST standardization and is considered as a promising candidate of this standardization.

Table 1 Security parameters for the QC-MDPC cryptosystem at 80, 128, and 256-bit security levels.

security	n_0	n	r	d_v	key size
80	2	9602	4801	45	4801
80	3	10779	3593	51	7186
80	4	12316	3079	55	9237
128	2	19714	9857	71	9857
128	3	22299	7433	81	14866
128	4	27212	6803	85	20409
256	2	65542	32771	137	32771
256	3	67593	22531	155	45062
256	4	81932	20483	161	61449

Table 2 BIKE parameters corresponding to the 128-bit security level.

security	n_0	n	r	d_v	key size
Level 1	2	24646	12323	71	12323

3. Key recovery attack

Our task is to obtain the full secret key $\boldsymbol{H}$. To this end, it is sufficient to know $\boldsymbol{H_0}$ because the remaining part of $\boldsymbol{H}$ can efficiently be determined by utilizing the generator matrix (public key) $\boldsymbol{G}$ in (2). Furthermore, due to the cyclicity of $\boldsymbol{H_0}$, we only need to recover its first row $\boldsymbol{h_0}$. Hereinafter, we identify $\boldsymbol{h_0}$ with the secret key itself.

Guo et al. proposed a reaction attack to recover $\boldsymbol{h_0}$,

which consists of two parts [8]. First, an attacker sends certain messages many times to the receiver and observes the decoding error rate (DER). The obtained statistical information allows us to construct a *distance spectrum* for the secret key $\boldsymbol{h_0}$, which is the set of distances between any two ones in the key. Then, recursive search using the distance spectrum recovers the key.

3.1 Distance spectrum

Consider a vector $\mathbf{c} = (c_0, \cdots, c_{r-1}) \in \mathbb{F}_2^r$ with $w(\boldsymbol{c}) = d_v$. We define the distance between c_i and c_j as follows:

$$d(i, j) = \min\{|i - j|, r - |i - j|\}. \tag{5}$$

Note that $\max_{i,j}(d(i, j)) = U := \lfloor r/2 \rfloor$. Accordingly, the distance multiplicities $\mu_{\boldsymbol{c}}(d)$ of d in the vector $\boldsymbol{c}$, and the distance spectrum $D(\boldsymbol{c})$, are defined as

$$\mu_{\boldsymbol{c}}(d) = \left|\{(i, j)|c_i = c_j = 1 \wedge d(i, j) = d\}\right|, \tag{6}$$

$$D(\boldsymbol{c}) = \{d|1 \le d \le U \wedge \mu_{\boldsymbol{c}}(d) > 0\}. \tag{7}$$

For later convenience, we also define the set of positions of ones in $\boldsymbol{c}$:

$$\begin{aligned}\boldsymbol{P}(\boldsymbol{c}) = &\left\{p_0, p_1, \cdots p_{d_v-1}\right\}, \text{ s.t.,}\\ &(\boldsymbol{c})_{p_i} = 1 \text{ for } 0 \le i \le d_v - 1.\end{aligned}$$

Hereinafter, we identify $\boldsymbol{P}(\boldsymbol{c})$ with $\boldsymbol{c}$ itself unless it causes confusion; for example, $D(\boldsymbol{P}(\boldsymbol{c}))$ simply denotes $D(\boldsymbol{c})$. One can calculate the above quantities for an example $\boldsymbol{c'} = (0100110)$ as follows:

$$\begin{aligned}D(\boldsymbol{c'}) =& \{1, 3\},\\ \mu_{c'}(1) = 1,\ \mu_{c'}(2) =& 0,\ \mu_{c'}(3) = 2,\\ \boldsymbol{P}(\boldsymbol{c'}) =& \{1, 4, 5\}.\end{aligned}$$

An attacker obtains the distance spectrum $D(\boldsymbol{h_0})$ through the following protocol. Hereinafter, we consider the case of $n_0 = 2$, in which the parity-check matrix is given by

$$\boldsymbol{H} = [\boldsymbol{H_0}\ \boldsymbol{H_1}]. \tag{8}$$

Let us define the set of error patterns Ψ_d as

$$\begin{aligned}\Psi_d = \{(\boldsymbol{a}, \boldsymbol{0}) \in \mathbb{F}_2^r \times \mathbb{F}_2^r | \exists \text{ distinct } s_1, s_2, \cdots, s_t, \text{ s.t.,}\\ \boldsymbol{P}(\boldsymbol{a}) = \{s_1, s_2, \cdots, s_t\}, \text{ and}\\ s_{2i} = (s_{2i-1} + d) \bmod r \text{ for } i = 1, \cdots, t/2\}.\end{aligned} \tag{9}$$

This set consists of words of length $2r$ with the first half containing $t/2$ pairs of ones at distance d, and the second half being the zero vector. For example, when $d = 2$, $t = 4$, $r = 10$, there is a vector $\boldsymbol{v} \in \Psi_d$:

$$\boldsymbol{v} = 01010010100000000000.$$

The attacker sends M messages with errors $\boldsymbol{v} \in \Psi_d$ to the receiver for each d. He counts the number of decoding errors and calculates an empirical DER. It is known that if d is included in the distance spectrum $D(\boldsymbol{h_0})$, the DER will be lower compared to when $d \notin D(\boldsymbol{h_0})$. Therefore, after sending messages $M \times U$ times, the attacker can expect which distances are included in $D(\boldsymbol{h_0})$.

The BF decoding and its variants are commonly utilized to decode QC-MDPC codes [7], [9]–[14]. These are, however, vulnerable to the above attack: when the receiver uses the BF decoding, the DER strongly depends on whether $d \in D(\boldsymbol{h_0})$ or not. To obtain the distance spectrum, we apply the BF decoding to sent messages $\boldsymbol{v} \in \Psi_d$, and simply calculate an empirical DER for each d as

$$\text{DER} = f/M,$$

where f is the number of decoding errors in M trials. Figure 1 depicts the DER for the QC-MDPC scheme with the parameters $(n, r, d_v, t) = (9602, 4801, 45, 110)$ for $1 \le d \le 600$, almost a quarter of U. In this experiment, we set $M = 10000$ for each d. As shown in Fig. 1, the DER decreases as the

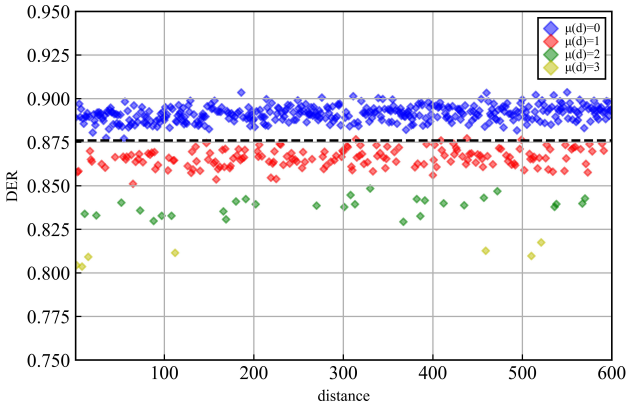


Fig. 1 The DER for the QC-MDPC scheme with parameters $(n, r, d_v, t) = (9602, 4801, 45, 110)$. For each d, the multiplicities $\mu(d) = 0, 1, 2, 3$ of the secret key are represented by blue, red, green, and yellow, respectively. We depict the threshold T with the dashed line.

corresponding multiplicity increases. Exploiting this observation, we determine a threshold T for the DER, which is represented by the dashed line in Fig. 1, and decide that distances d with the DER below T are in the distance spectrum of the secret key. The algorithm for determining the distance spectrum using the obtained threshold T is outlined below.

Algorithm 1 Compute the spectrum

Input: threshold T, number of decoding trials M per distance, upper distance U.
Output: $D(\boldsymbol{h_0})$.

```
D(h_0) ← {}.
for all i (1 ≤ i ≤ U) do
    f ← 0.
    for all j (0 ≤ j < M) do
        take v ∈ Ψ_i.
```

6: Apply BF decoding to $\boldsymbol{v}$.
7: **if** $\boldsymbol{H}\boldsymbol{v}^T \neq \boldsymbol{0}$ **then**
8: $f \leftarrow f + 1$.
9: **end if**
10: **end for**
11: **if** $f/M \leq T$ **then**
12: $D(\boldsymbol{h_0}) \leftarrow D(\boldsymbol{h_0}) \cup \{i\}$.
13: **end if**
14: **end for**
15: **return** $D(\boldsymbol{h_0})$.

The above protocol typically requires the number of messages $M \sim 1/\mathrm{DER}$. Thus, when we try to achieve the λ-bit security level, the DER must be in the order of $2^{-\lambda}$. One way to realize such a DER is sufficiently decreasing the error weight t. Actually, BIKE adopts so small t that the DER will be in the order of 2^{-128}. In near future, however, such a small t may provide vulnerabilities to other attacks, such as information set decoding [21]. Thus, this simple way of decreasing the error weight will be too naive to neutralize the above reaction-based attack. Utilizing other methods than the BF decoding and its variants, such as an ADMM-based one [22], can be another way to reduce the DER.

It is also worth mentioning that the reaction-based attack is not the only way to obtain the distance spectrum. For example, Ref. [23] considers a timing attack that exploits a correlation between distances in the secret key and the number of iterations required for decoding. Only taking measures against the reaction-based attack may not be sufficient.

3.2 Reconstruction

The reconstruction of $\boldsymbol{h_0}$ from the distance spectrum was also proposed in [8], which uses recursive search to find a candidate of $\boldsymbol{h_0}$. Let us align the distance spectrum $D(\boldsymbol{h_0})$ in increasing order:

$$D(\boldsymbol{h_0}) = \{i_0, i_1, \cdots\}. \tag{10}$$

We name a candidate of the secret key as $\boldsymbol{h_0'}$ with

$$\boldsymbol{P}(\boldsymbol{h_0'}) = (p_0, p_1, \cdots, p_{d_v-1}). \tag{11}$$

We can fix $p_0 = 0$ and $p_1 = i_0$ without loss of generality. Note that what we will reproduce is not necessarily the secret key itself. It can be a cyclic shift or a mirror image of the key. See Sec. 4 for the details.

First, tentatively assign $p_1 + i_0$ to p_2. Then, check whether the distance between p_0 and p_2 exists in $D(\boldsymbol{h_0})$. If it does, update $p_2 = p_1 + i_0$. If not, assign $p_1 + i_1$ to p_2. Again, check whether the distance between p_0 and p_2 exists in $D(\boldsymbol{h_0})$. Repeat this process until reaching p_{d_v-1}. This reconstruction algorithm is summarized in Algorithm 2.

Algorithm 2 Key recovery using recursive search [8]

Input: $D(\boldsymbol{h_0})$, $p_0 = 0$, $p_1 = \min\left(D(\boldsymbol{h_0})\right)$, $l = 2$
Output: secret key $\boldsymbol{h_0}$
1: **for all** i $(p_{l-1} + 1 \leq i \leq r)$ **do**
2: **for all** j $(0 \leq j \leq l - 1)$ **do**
3: **if** $i - p_j \in D(\boldsymbol{h_0})$ **then**
4: $p_l \leftarrow i$.
5: $l \leftarrow l + 1$.
6: **else**
7: goto line 20
8: **end if**
9: **end for**
10: **if** $l = dv$ **then**
11: **if** $D(\boldsymbol{h_0'}) = D(\boldsymbol{h_0})$ **then**
12: **return** $\boldsymbol{h_0}$.
13: **else**
14: **return**
15: **end if**
16: **end if**
17: Recursive call with $D(\boldsymbol{h_0}), l$
18: $l \leftarrow l - 1$.
19: $p_l \leftarrow 0$.
20: **return**
21: **end for**
22: **return** Failure to recover secret key

As this algorithm involves recursion (STEP 17), the reconstruction of $\boldsymbol{h_0}$ will take considerable time in general. In what follows, we propose another method to decrease the reconstruction time of $\boldsymbol{h_0}$.

4. Proposed methods

We interpret the key reconstruction from the distance spectrum into a clique problem, as suggested by Fabšič et al [16]. Let us introduce the clique problem. A graph G is represented as $G = (V, E)$, where V is the set of vertices and E is the set of edges. In graph theory, a clique $C = (V', E')$ in G is a subgraph where every two vertices in V' are connected by an edge in E'. The size of a clique refers to the number of vertices belonging to that clique. The clique problem is to determine whether there is a clique of a given size in a given graph.

We define a graph $G = (V, E)$ whose vertices are elements of

$$V = \overline{D(\boldsymbol{h_0})} := \{0\} \cup D(\boldsymbol{h_0}) \cup (r - D(\boldsymbol{h_0})), \tag{12}$$

where $r - D(\boldsymbol{h_0})$ is the set obtained by subtracting each element of $D(\boldsymbol{h_0})$ from r. The set of edges E is constructed by the following rule:

$$E := \{(u, v) \mid\ d(v, u) \in D(\boldsymbol{h_0}),\ v, u \in \overline{D(\boldsymbol{h_0})}\}. \tag{13}$$

The set of vertices (12) corresponds to all possible positions of ones when we fix $p_0 = 0$. if we consider a cyclic shift or a mirror image of the true key, which satisfies $p_0 = 0$, and represent it by the same symbol $\boldsymbol{h_0}$, $\boldsymbol{P}(\boldsymbol{h_0})$ is a subset of $\overline{D(\boldsymbol{h_0})}$ by definition. Moreover, a distance between any two elements in $\boldsymbol{P}(\boldsymbol{h_0})$ is in the distance spectrum, i.e.,

$$d(p_i, p_j) \in D(\boldsymbol{h_0}), \ \ \forall p_{i,j} \in \boldsymbol{P}(\boldsymbol{h_0}).$$

Summarizing, the positions of ones of the secret key is represented in G by vertices in $\overline{D(\boldsymbol{h_0})}$, any two of which are connected by a edge in E. In other words, the secret key is a d_v-clique in G. Thus, our task is regarded as search for a d_v-clique (clique of size d_v) corresponding to the secret key, which is known to be NP-hard. Fabšič et al. applied this observation to reconstruct the secret key in the QC-LDPC scheme. In their case, searching for a clique with a much smaller size than d_v is sufficient: due to sparsity of secret keys in QC-LDPC codes, such a small clique almost uniquely determines a d_v-clique that contains the small one.

When considering the QC-MDPC scheme, we need to find a larger clique than in the case of the QC-LDPC scheme. To accomplish this, we employ breadth-first search. The specific algorithm is summarized in Algorithm 3.

Algorithm 3 Proposed algorithm

Input: $D(\boldsymbol{h_0})$, $p_0 = 0$, $p_1 = \min(D(\boldsymbol{h_0}))$, $r = |\boldsymbol{h_0}|$
Output: set of secret key candidates $\mathcal{H}_{\boldsymbol{0}}$

$\mathcal{H}_{\boldsymbol{0}} \leftarrow \{\}$.
$\overline{D(\boldsymbol{h_0})} = \{p_0\} \cup D(\boldsymbol{h_0}) \cup (r - D(\boldsymbol{h_0}))$.
$S_0 \leftarrow \{p_0, p_1\}$.
$S_0^{'} \leftarrow \{j \in \overline{D(\boldsymbol{h_0})} \setminus S_0 \mid d(j, p_1) \in D(\boldsymbol{h_0})\}$.
$N \leftarrow 1, t \leftarrow 2$.
$\mathcal{S} = \{S_0, S_1, \cdots, S_{N-1}\}$.
$\mathcal{S}^{'} = \{S_0^{'}, S_1^{'}, \cdots, S_{N-1}^{'}\}$.
while $t < d_v$ **do**
  $\tilde{\mathcal{S}} \leftarrow \{\}, \tilde{\mathcal{S}}' \leftarrow \{\}$.
  for all i $(0 \le i < N)$ **do**
    $S_i^{'} = \{a_0, a_1, \cdots\}$.
    for all l $(0 \le l < |S_i^{'}|)$ **do**
      $S_{i,l} \leftarrow S_i \cup \{a_l\}$.
      $S'_{i,l} \leftarrow \{q \in S_i' \setminus \{a_l\} \mid q > a_l, d(q, a_l) \in D(\boldsymbol{h_0})\}$.
      $B_{i,l} \leftarrow S_{i,l} \cup S'_{i,l}$.
      if $|B_{i,l}| = d_v$ **and** $D(\boldsymbol{h_0}) = D(B_{i,l})$ **then**
        append $B_{i,l}$ to $\mathcal{H}_{\boldsymbol{0}}$.
      end if
      if $|B_{i,l}| > d_v$ **then**
        append $S_{i,l}$ to $\tilde{\mathcal{S}}$, and $S'_{i,l}$ to $\tilde{\mathcal{S}}'$.
      end if
    end for
  end for
  $\{\mathcal{S}, \mathcal{S}^{'}\} \leftarrow \{\tilde{\mathcal{S}}, \tilde{\mathcal{S}}'\}$.
  $N \leftarrow |\mathcal{S}|, t \leftarrow t + 1$.
end while
for all i $(0 \le i < |N|)$ **do**
  if $D(\boldsymbol{h_0}) = D(S_i)$ **then**
    append S_i to $\mathcal{H}_{\boldsymbol{0}}$.
  end if
end for
return $\mathcal{H}_{\boldsymbol{0}}$

We provide a formal explanation for the algorithm in terms of the graph G, whose vertices are the elements of $\overline{D(\boldsymbol{h_0})}$. At the t-th step, we know the set of all t-cliques $\mathcal{S}^{(t)} = (S_0^{(t)}, S_1^{(t)}, \cdots)$, where each $S_i^{(t)}$ represents a t-clique. Accordingly, we have $\mathcal{S}'^{(t)} = (S_0'^{(t)}, S_1^{'(t)}, \cdots)$, in which $S_i'^{(t)}$ is the set of the vertices connected to all elements of the clique $S_i^{(t)}$. Taking an element $a_l \in S_i'^{(t)}$, we create the $(t+1)$-clique and its complementary set as

$$\begin{aligned} S_{i,l}^{(t+1)} &:= S_i^{(t)} \cup \{a_l\}, \\ S_{i,l}'^{(t+1)} &:= \{q \in S_i'^{(t)} \setminus \{a_l\} \mid q > a_l, d(q, a_l) \in D(\boldsymbol{h_0})\}. \end{aligned} \tag{14}$$

We then evaluate the size of their union,

$$B_{i,l}^{(t+1)} := S_{i,l}^{(t+1)} \cup S_{i,l}'^{(t+1)}.$$

If its size is less than d_v, this set never contains d_v cliques, or equivalently, candidates of the secret key. Hence, we discard $B_{i,l}^{(t+1)}$. If not, this union can contains candidates of the key and thus we bring $S_{i,l}^{(t+1)}$ and $S_{i,l}'^{(t+1)}$ to the $(t+1)$-th step while relabeling them by a new index i' instead of (i, l). In particular, if its size equals to d_v, the union can be a candidate of the key. We check whether $B_{i,l}^{(t+1)}$ meets the condition $D(\boldsymbol{h_0}) = D(B_{i,l}^{(t+1)})$, and if it does, we append it to $\mathcal{H}_{\boldsymbol{0}}$, a set of secret key candidates. In the case of the QC-LDPC scheme, e.g., with the parameter $(r, d_v) = (4096, 13)$, a small $t \sim 3, 4$ is found to be sufficient [16]: the sparsity of the secret key rapidly reduces the size of $S_i'^{(t)}$ while t increases. The number of steps required for the QC-MDPC scheme is around 9 or 10 according to our experiment.

Let us show a simple example in which the secret key is $\boldsymbol{h_0} = 011010010000$, $r = 12$, and $d_v = 4$. The distance spectrum $D(\boldsymbol{h_0})$ is given by

$$D(\boldsymbol{h_0}) = \{1, 2, 3, 5, 6\}.$$

Our task is to enumerate secret key candidates $\boldsymbol{h_0'}$ with

$$\boldsymbol{P}(\boldsymbol{h_0'}) = (p_0, p_1, p_2, p_3), \tag{15}$$

from the above distance spectrum. We initialize $p_0 = 0$ and $p_1 = \min(D(\boldsymbol{h_0})) = 1$. $\overline{D(\boldsymbol{h_0})}$ in this example is

$$\begin{aligned} \overline{D(\boldsymbol{h_0})} &= \{0\} \cup D(\boldsymbol{h_0}) \cup (r - D(\boldsymbol{h_0})) \\ &= \{0, 1, 2, 3, 5, 6, 7, 9, 10, 11\}. \end{aligned}$$

The associated graph G, in which each vertex corresponds to an element in $\overline{D(\boldsymbol{h_0})}$, is shown in Fig. 2. The set of edges are determined by Eq. (13). We begin with $S_0 = \{p_0, p_1\} = \{0, 1\}$ and evaluate the compliment,

$$S_0^{'} = \{2, 3, 6, 7, 10, 11\}.$$

Following Eq. (14), we then construct $S_{i,l}$, and $S'_{i,l}$, that is,

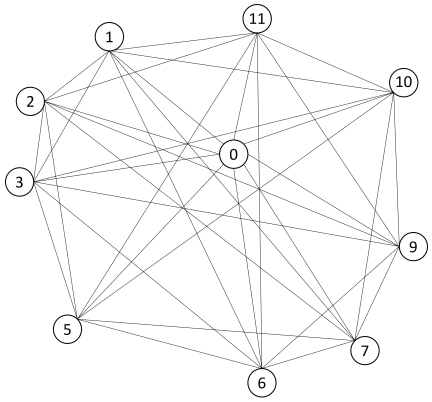


Fig. 2 Graph G constructed from the distance spectrum. Each vertex corresponds to an element of $\overline{D(\boldsymbol{h_0})}$ while an edge is drawn when the distance between the corresponding endpoints is in the spectrum.

$$
\begin{aligned}
S_{0,0} &= \{0,1,2\}\,,\ S'_{0,0} = \{3,7,11\},\\
S_{0,1} &= \{0,1,3\}\,,\ S'_{0,1} = \{6,10\},\\
S_{0,2} &= \{0,1,6\}\,,\ S'_{0,2} = \{7,11\},\\
S_{0,3} &= \{0,1,7\}\,,\ S'_{0,3} = \{10\},\\
S_{0,4} &= \{0,1,10\}\,,\ S'_{0,4} = \{11\},\\
S_{0,5} &= \{0,1,11\}\,,\ S'_{0,5} = \{\,\}.
\end{aligned}
$$

One can see that when the indices (i,l) are $(0,3)$ or $(0,4)$, the size of their union $B_{i,l}$ is $d_v = 4$. We confirm whether the distance spectrum of either set matches $D(\boldsymbol{h_0})$ (STEP 16), and found that,

$$B_{0,3} = \{0,1,7,10\}\,, \tag{16}$$

meets the condition. Therefore, we append this into $\boldsymbol{\hbar_0}$ as a secret key candidate. Figure 3 shows the graph and the clique corresponding to $B_{0,3}$. As the size of the union $B_{0,5}$

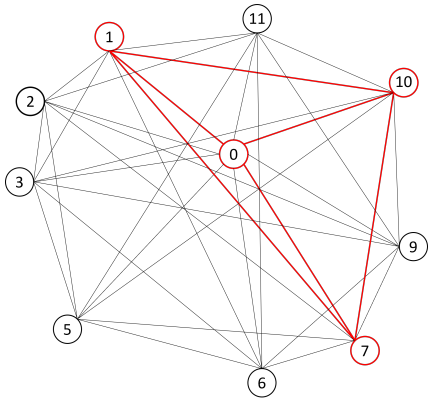


Fig. 3 Graph G with the 4-clique corresponding to $B_{0,3} = \{0,1,7,10\}$. The clique is depicted in red.

is less than 4, $S_{0,5}$ and $S'_{0,5}$ are discarded. For the other indices, that is, $(i,l) = (0,0)$, $(0,1)$ and $(0,2)$, the size of $B_{i,l}$ is greater than 4, and thus we add $S_{i,l}$ and $S'_{i,l}$ to $\mathcal{S}$ and $\mathcal{S}'$ respectively (STEP 20 and 24) as

$$
\begin{aligned}
\mathcal{S} &= \{\{0,1,2\},\{0,1,3\},\{0,1,6\}\}\,,\\
\mathcal{S}' &= \{\{3,7,11\},\{6,10\},\{7,11\}\}\,,
\end{aligned}
$$

with renumbering the elements by a new index $i = 0,1,2$. The explicit forms of the elements are

$$
\begin{aligned}
S_0 &= \{0,1,2\}\,,\ S'_0 = \{3,7,11\},\\
S_1 &= \{0,1,3\}\,,\ S'_1 = \{6,10\},\\
S_2 &= \{0,1,6\}\,,\ S'_2 = \{7,11\}.
\end{aligned}
$$

Proceed to the next step. Similarly to the previous step, we obtain

$$
\begin{aligned}
S_{0,0} &= \{0,1,2,3\}\,,\ S'_{0,0} = \{\,\},\\
S_{0,1} &= \{0,1,2,7\}\,,\ S'_{0,1} = \{\,\},\\
S_{0,2} &= \{0,1,2,11\}\,,\ S'_{0,2} = \{\,\},\\
S_{1,0} &= \{0,1,3,6\}\,,\ S'_{1,0} = \{\,\},\\
S_{1,1} &= \{0,1,3,10\}\,,\ S'_{1,1} = \{\,\},\\
S_{2,0} &= \{0,1,6,7\}\,,\ S'_{2,0} = \{\,\},\\
S_{2,1} &= \{0,1,6,11\}\,,\ S'_{2,1} = \{\,\}.
\end{aligned}
$$

Now, the size of the cliques reaches $d_v = 4$, and thus the search is terminated. Checking whether each clique meets the condition $D(\boldsymbol{h_0}) = D(S_{i,l})$, we found that

$$S_{1,0} = \{0,1,3,6\}, \tag{17}$$

is the only remaining candidate of the key, which is represented by the graph shown in Fig. 4.

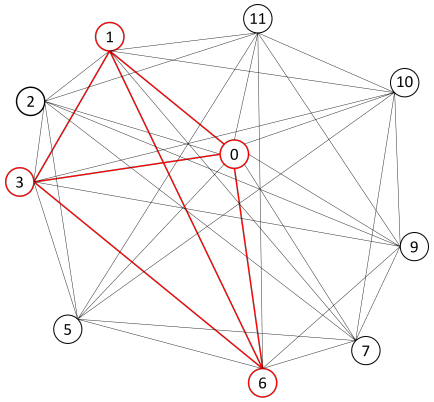


Fig. 4 G with the 4-clique corresponding to $S_{1,0} = \{0,1,3,6\}$. The clique is depicted in red.

The bit sequences corresponding to Eqs. (16) and (17) are denoted by $\boldsymbol{h}'_{\boldsymbol{0}}$ and $\widetilde{\boldsymbol{h}'_{\boldsymbol{0}}}$, respectively, and they are explicitly represented by

$$\boldsymbol{h}'_{\boldsymbol{0}} = 110000010010, \tag{18}$$

$$\widetilde{\boldsymbol{h}'_{\boldsymbol{0}}} = 110100100000. \tag{19}$$

Note that the latter candidate is the mirror image of the former with respect to p_1. Also, the true secret key $\boldsymbol{h_0}$ is obtained by right-shifting the latter by 1 bit. In general, candidates after the reconstruction is not $\boldsymbol{h_0}$ itself: the distance spectrum $D(\boldsymbol{h_0})$ is invariant under cyclic shift and mirroring, which introduces ambiguity to the reconstruction results. In other words, an obtained candidate is a row in $\boldsymbol{H_0}$ or its mirror image, but not necessarily the first row. To reproduce the true parity-check matrix, we generate $2r$ candidate matrices through rotation and mirroring of the obtained candidates and take a matrix that can successfully decrypt encoded messages.

We should also mention that the uniqueness of candidates reconstructed from a distance spectrum is not guaranteed in general. For example, when $D(\boldsymbol{h_0}) = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $r = 17$, $d_v = 6$, there are two non-trivially different candidates:

$$\begin{aligned} \boldsymbol{h_0'} &= 11011010100000000, \\ \boldsymbol{h_0'} &= 11010110100000000, \end{aligned} \tag{20}$$

where the latter is neither a circular shift nor a mirror image of the former. The possibility of this ambiguity for realistic situations is discussed in Sec. 5.

5. Numerical Experiments

We compare the reconstruction time of our algorithm with the recursive search proposed by Guo et al. while varying the block length r. Let the weight w of the parity-check matrix scale in $O(\sqrt{r \log r})$ [7]. We interpolate several parameters (r, d_v) between the ones for 80-bit and 128-bit security with $n_0 = 2$ in Table 1. Assuming the dependence $a\sqrt{r \log r} + b = w$ with $a, b \in \mathbb{R}$, we obtain $w = 0.523\sqrt{r \log r} - 15.6$.

Table 3 presents the reconstruction time of $\boldsymbol{h_0}$ with varying (r, d_v) according to the above dependence. The execution time is limited to 7 days (approximately 600,000 seconds), and reconstruction trials for each parameter are conducted 10 times with randomly generated secret keys. The experimental results indicate that our method significantly reduces the reconstruction time comparing to the recursive search. For $r \geq 9000$, the recursive search sometimes exceed the time limit, whereas the reconstruction by our method does never exceed the limit for any parameters in the experiment. The results were obtained using a 12th Gen Intel(R) Core(TM) i9-12900KF.

The recursive search exhibits significant dispersion in reconstruction times. This is attributed to the depth-first search approach used in the algorithm: it is uncertain when we reach the solution. Therefore, if the solution is found quickly, the reconstruction is completed promptly, otherwise, it may take considerable time. On the other hand, the reconstruction using our method shows less dispersion because it employs a breadth-first search approach.

As our method is breadth-fist search, the high time performance is realized at the cost of its space complexity. In our experiments, we save the sets of current cliques $\mathcal{S}^{(t)}$ and their compliments $\mathcal{S}'^{(t)}$ in storage at t-th step of the algorithm, and update them, whereas the recursive search only stores a current key candidate in memory. The file size for the largest parameter set $(n, r, d_v) = (19714, 9857, 71)$ reaches the maximum $\sim$ 3 GiB at $t = 4 \sim 6$. Meanwhile, actual memory consumption in our algorithm is comparable with that in the recursive search $\sim$ 800 MiB, thanks to the usage of storage.

We should also mention that the recursive search by Guo et al. halts once a candidate of the secret key is found. As aforementioned, the uniqueness of reconstructed key candidates (up to shift and mirror images) is not guaranteed in general. If the ambiguity of candidates occurs for realistic parameters, the recursive search must be exhaustive, and it takes even more time than in Table 3. The reconstruction time for exhaustive search is shown in Table 4. The secret key is randomly picked out of those used in the experiment of Table 3. The exhaustive search takes much more time than the corresponding one-candidate search as expected. On the other hand, our method is exhaustive search by definition (see Algorithm 3), and thus it takes no additional time even if there exist several non-trivially different candidates.

Table 3 Reconstruction time (in seconds) of $\boldsymbol{h_0}$ using the recursive search (Algorithm 2) and our method (Algorithm 3). In each cell, the shortest, average, and longest reconstruction times in 10 experiments are shown from top to bottom. The maximum execution time is set to 7 days.

(n, r, d_v)	recursive search (Algorithm 2)	our method (Algorithm 3)
(2000, 1000, 15)	0.082 0.084 0.087	0.081 0.084 0.088
(4000, 2000, 25)	0.275 0.299 0.316	0.297 0.305 0.312
(6000, 3000, 33)	0.596 1.03 1.75	0.635 0.675 0.717
(8000, 4000, 39)	1.11 3.59 7.51	1.11 1.19 1.29
(9602, 4801, 45)	3.26 837 7998	1.78 1.93 2.16
(10000, 5000, 47)	4.47 22.5 66.0	2.17 2.41 2.88
(12000, 6000, 51)	3.61 185 599	3.39 4.22 6.10
(14000, 7000, 57)	140 4125 21805	5.92 11.1 16.6
(16000, 8000, 63)	1625 73053 325364	22.8 48.3 111
(18000, 9000, 67)	1318 - >7days	69.2 98.0 127
(19714, 9857, 71)	110813 - >7days	106 177 247

Table 4 The reconstruction time (in seconds) for the exhaustive search by Guo et al. The maximum execution time is set to 7 days.

(n, r, d_v)	exhaustive search
(2000, 1000, 13)	0.082
(4000, 2000, 23)	0.876
(6000, 3000, 33)	742
(8000, 4000, 39)	44930
(9602, 4801, 45)	>7days

We, however, expect that in realistic situations, such ambiguity may not occur; in our experiments, we have never observed non-trivially different solutions with a same distance spectrum. This may be thanks to sparsity of secret keys in QC-MDPC codes. The secret key in Eq. (20) has a high density of non-zero elements, that is, $d_v = 6$ to $r = 17$, while in this paper, the parameters for 80-bit security with $n_0 = 2$, have a low density, $d_v = 45$ to $r = 4801$. Although one can guess that sparsity will strongly restrict the number of possible (non-trivially different) solutions, clarifying uniqueness conditions is still an open problem.

The reconstruction time for the Level 1 BIKE parameters (Table 2) is shown in Table 5. The results for our methods show that parity-check matrices with the BIKE parameters is easier to reconstruct than those in Table 3. This will be due to the BIKE parameters being sparser than the parameters in Table 3.

Table 5 Reconstruction time (in seconds) using the recursive search (Algorithm 2) and our method (Algorithm 3) with the Level 1 BIKE parameters. The shortest, average, and longest reconstruction times in 10 experiments are shown from top to bottom.

(n, r, d_v)	recursive search (Algorithm 2)	our method (Algorithm 3)
(24646, 12323, 71)	2290	17.1
	53311	20.2
	222390	24.6

6. Conclusion

In this paper, we have proposed a method to reconstruct a secret key of the QC-MDPC McEliece cryptsystem. The original algorithm proposed by Guo et al. uses recursive search in key reconstruction from the distance spectrum [8]. We, instead, exploit breadth-first search to reconstruct the key, motivated by the interpretation of the problem into a clique problem, which is indicated in [16]. We found that the key reconstruction time by our method is much shorter than that by the recursive search. Moreover, the dispersion in the reconstruction time is less in our method.

It is worth noting that our method can accomplish the reconstruction within a realistic time scale even if the distance spectrum has several non-trivially different keys. This is not the case for the recursive search: exhaustive recursive search obviously takes much more time than one-candidate search. As the reconstruction from a distance spectrum does not necessarily have a unique solution, this property in our method can be advantageous over the recursive search. However, we should also annotate that we have never observed non-trivially different solutions with a same distance within our experiments. We expect that sparsity of secret keys in QC-MDPC codes will ensure the uniqueness with a high probability, but this is still obscure. It will be important to delve into this uniqueness problem.

We have focused in this paper on the time complexity of key reconstruction (the second part of the attack) from the distance spectrum. We here mention the time complexity of acquisition of the distance spectrum (the first part of the attack). According to our experiment for the parameters $(n, r, d_v, t) = (9602, 4801, 45, 110)$ and $M = 1000000$, the first part approximately takes a month, which will be longer than the second typically does. Hence, improving the time complexity of the second part will have a restrictive impact on effectiveness of the attack for now. It is, however, unclear that the relationship between their time complexities holds even when we change the parameters or the decoding algorithm. In particular, one can naively expect that when n increases, the complexity of the first part will grow polynomially while that of the second will grow exponentially. It is critical when the complexity of the second part dominates, and therefore we leave the detailed discussion for future work.

We also note that Paiva and Terada proposed to utilize the distance spectrum of the remaining parts in the parity-check matrix [17]. They showed that this additional information accelerates the key reconstruction. Comparing our method with theirs in realistic situations is also our future work.

Acknowledgments

This work was supported by JSPS Grants-in-Aid for Scientific Research Grant Number JP18K03387.

References

[1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, pp.124–134, 1994.

[2] "Post-Quantum Cryptography." Available: `https://csrc.nist.gov/Projects/post-quantum-cryptography/`.

[3] R. McEliece, "A public key cryptosystem based on algebraic coding theory," DSN Prog. Re., vol.42-44, pp.114–116, 1978.

[4] T.P. Berger, P.L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key Length of the McEliece Cryptosystem," International Conference on Cryptology in Africa, pp.77–97, Springer, 2009.

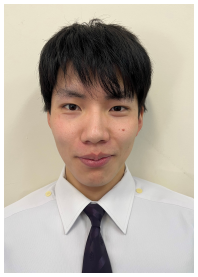
[5] M. Repka and P. Zajac, "Overview of the McEliece cryptosystem and its security," Tatra Mountains Mathematical Publications, vol.60, no.1, pp.57–83, 2014.

[6] J. Bolkema, H. Gluesing-Luerssen, C.A. Kelley, K.E. Lauter, B. Malmskog, and J. Rosenthal, "Variations of the McEliece Cryptosystem," Algebraic Geometry for Coding Theory and Cryptography: IPAM, Los Angeles, CA, February 2016, pp.129–150, Springer, 2017.

[7] R. Misoczki, J.P. Tillich, N. Sendrier, and P.S. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," 2013 IEEE international symposium on information

theory, pp.2069–2073, IEEE, 2013.
[8] Q. Guo, T. Johansson, and P.S. Wagner, "A key recovery reaction attack on QC-MDPC," IEEE Transactions on Information Theory, vol.65, no.3, pp.1845–1861, 2019.
[9] I.E. Bocharova, T. Johansson, and B.D. Kudryashov, "Improved iterative decoding of QC-MDPC codes in the McEliece public key cryptosystem," 2019 IEEE International Symposium on Information Theory (ISIT), pp.1882–1886, IEEE, 2019.
[10] N. Drucker, S. Gueron, and D. Kostic, "QC-MDPC Decoders with Several Shades of Gray," International Conference on Post-Quantum Cryptography, pp.35–50, Springer, 2020.
[11] H. Kaneko, "Look-Ahead Bit-Flipping Decoding of MDPC Code," 2022 IEEE International Symposium on Information Theory (ISIT), pp.2922–2927, IEEE, 2022.
[12] A. Nilsson, I.E. Bocharova, B.D. Kudryashov, and T. Johansson, "A Weighted Bit Flipping Decoder for QC-MDPC-based Cryptosystems," 2021 IEEE International Symposium on Information Theory (ISIT), pp.1266–1271, IEEE, 2021.
[13] N. Sendrier and V. Vasseur, "About low DFR for QC-MDPC decoding," International Conference on Post-Quantum Cryptography, pp.20–34, Springer, 2020.
[14] "QC-MDPC decoder." `https://github.com/vvasseur/qcmdpc\_decoder`.
[15] H. Bartz and G. Liva, "On decoding schemes for the MDPC-McEliece cryptosystem," SCC 2019; 12th International ITG Conference on Systems, Communications and Coding, pp.1–6, VDE, 2019.
[16] T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson, "A reaction attack on the QC-LDPC McEliece cryptosystem," Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8, pp.51–68, Springer, 2017.
[17] T.B. Paiva and R. Terada, "Improving the efficiency of a reaction attack on the QC-MDPC McEliece," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.101, no.10, pp.1676–1686, 2018.
[18] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," IEEE Transactions on Information Theory, vol.24, no.3, pp.384–386, 1978.
[19] K. Kurosawa and K. Inaba, "Consideration on the np completeness of linear codes," The transactions of the Institute of Electronics, Information and Communication Engineers. A, vol.J68-A, no.9, pp.953–956, 1985.
[20] "*BIKE (Bit Flipping Key Encapsulation)*." Available: `https://bikesuite.org/` (2024/06/05).
[21] S. Narisada, K. Fukushima, and S. Kiyomoto, "Multiparallel MMT: Faster ISD Algorithm Solving High-Dimensional Syndrome Decoding Problem," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E106.A, no.3, pp.241–252, 2023.
[22] K. Watanabe, M. Ohtsuka, and Y. Tsukie, "ADMM and Reproducing Sum-Product Decoding Algorithm Applied to QC-MDPC Code-Based McEliece Cryptosystems," IEEE Transactions on Information Theory, vol.70, no.3, pp.1774–1786, 2024.
[23] E. Eaton, M. Lequesne, A. Parent, and N. Sendrier, "QC-MDPC: A Timing Attack and a CCA2 KEM," Post-Quantum Cryptography, ed. T. Lange and R. Steinwandt, Cham, pp.47–76, Springer International Publishing, 2018.

Motonari OHTSUKA was born in Nara prefecture, Japan in 1996. He received the B.S. degree in computer science from National Defense Academy of Japan in 2019. He is currently pursuing M.S. degree in mathematics and computer science at National Defense Academy of Japan.



Takahiro ISHIMARU was born in Fukuoka prefecture, Japan in 1997. He received the B.S. degree in mathematics from National Defense Academy of Japan in 2019. He is currently pursuing M.S. degree in mathematics and computer science at National Defense Academy of Japan.

Yuta TSUKIE was born in Saitama prefecture, Japan in 1988. He received the B.S. degree in electrical engineering from Tokyo University of Science in 2017 and he received M.S. degree in mathematics and computer science at National Defense Academy of Japan in 2023.

Shingo KUKITA received the B. S., M.S., and Ph.D. degrees in physics from Nagoya University in 2012, 2014, 2018, respectively. Since 2023 he has been an Assistant Professor in Department of Computer Science, National Defense Academy of Japan. His current research interests are in the area of quantum control, open quantum system, and quantum information theory.

Kohtaro WATANABE was born in Kanagawa prefecture, Japan in 1965. He received the B.S., M.S. and Ph.D. degrees in mathematics from Tokyo Institute of Technology in 1989, 1991 and 2004 respectively. Since 2014, he has been a Professor of Department of Computer Science, National Defense Academy of Japan. His research interests includes non-linear ordinary and partial differential equations and information theory.