A proposal for quantum GRS algorithm and the cryptanalysis for ROLLO and RQC

Asuka Wakasugi[1] and Mitsuru Tada[2]

[1] Graduate School of Science and Engineering, Chiba University, Japan*
a_wakasugi@eaglys.co.jp
[2] Graduate School of Science, Chiba University, Japan
m.tada@faculty.chiba-u.jp

26 June 2023

Abstract. Code-Based Cryptosystem, CBC, is one of the candidates for Post-Quantum Cryptosystems, PQCs. Its security primarily bases on the Syndrome Decoding Problem, SDP. In this paper, we focus on the rank CBC whose security relies on the rank SDP. The GRS (Gaborit-Ruatta-Schrek) algorithm is well known as the current best decoding algorithm for the rank SDP. We propose the quantum version of the GRS algorithm. Then, we introduce the attack strategy using that quantum algorithm for previous rank CBCs remained at the 2nd Round of the NIST's PQC standardization project, and consider the quantum security for those cryptosystems. We present a result that is effective for RQC by our attack method, so give new RQC's instances which is secure against that attack.

Keywords: Rank code-based cryptography · GRS algorithm · Grover's algorithm · ROLLO · RQC

1 Introduction

A public-key cryptosystem is currently used for our secure communication. Its security largely bases on number theory problems such as the integer factorization problem or the discrete logarithm problem. The RSA cryptosystem relies on the former problem. In 1994, Shor [25] showed polynomial-time quantum algorithms to solve both problems, and therefore once a quantum computer of an appropriate size is built, the RSA cryptosystems shall be vulnerable to the Shor's algorithm. Then, we need to think about Post-Quantum Cryptosystems, PQCs, and code-based cryptosystem, CBC, is among the promising candidates for PQC.

1.1 Rank-metric code

Let q be a prime power, and m and n be positive integers. For $x \in \mathbb{F}_{q^m}^n$, let φ_x be the linear transformation such that $\varphi_x : \mathbb{F}_q^n \ni v \mapsto vx \in \mathbb{F}_{q^m}$, and $\mathrm{rank}_{\mathbb{F}_q}(x)$

* Now with EAGLYS Inc, Research and Development, Tokyo, Japan.

denotes the rank of x, i.e. the dimension of the image of φ_x. Also, an $\mathbb{F}_{q^m}$-linear code $\mathcal{C}$ of n-length and k-dimension is a sublinear space with k-dimension in $\mathbb{F}_{q^m}^n$. Such a $\mathcal{C}$ is called an $[n,k]_{q^m}$-linear code. A rank-metric code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ is an $[n,k]_{q^m}$-linear code with rank-metric. For example, the Gabidulin code [5] and the LRPC code [6] are rank-metric codes. These codes correspond to the rank version of the Reed-Solomon code [23] and the LDPC code [9] with Hamming metric, respectively.

1.2 Rank Syndrome Decoding Problem(Rank SDP)

Definition 1 (Rank SDP). *Let q, m, n, k and w be positive integers, H be a matrix in $\mathbb{F}_{q^m}^{(n-k)\times n}$, and s be a vector in $\mathbb{F}_{q^m}^{n-k}$. Then, the rank SDP is the problem to find an $e \in \mathbb{F}_{q^m}^n$ such that $He^T = s$ and $\mathrm{rank}_{\mathbb{F}_{q^m}}(e) = w$, on input q, m, n, k, w, H and s.*

The rank SDP is not known to be NP-complete, but it is known that there is a UR reduction [12] between the rank SDP and the SDP with Hamming-metric, which is NP-complete [8]. In this paper, we consider the rank code-based cryptosystem, rank CBC, which uses the rank SDP as the basis for its security. NIST has been standardrizing PQCs since 2016, and the project is currently in its 4th Round. Two rank CBCs, ROLLO and RQC, remained in PQC standardization project until its 2nd Round, but do not remain in its 4th Round.

1.3 Attack method for rank CBC

The rank CBC is the McEliece cryptosystem [16] using the rank-metric code. The GRS algorithm [7] is the best known decoding algorithm to solve the rank SDP. Also, the algebraic attack is one of the attack methods which solve the equations with multi-variate polynomials derived from the public key in the multi-variate polynomial based public-key cryptosystem, and can be applied to the rank CBC. Bardet et al. [2] proposed the algebraic attack for ROLLO and RQC. In this paper, we focus on the GRS algorithm as the previous attack method for the rank CBC. We review the abstract of the GRS algorithm in Section 2.

1.4 Previous study

CBC is derived from the McEliece cryptosystem. Since the McEliece cryptosystem appeared in 1978, there are some studies of security from a classical view. For example, the Information Set Decoding(ISD) algorithm is the algorithm to solve the Syndrome Decoding Problem with Hamming-metric. Prange [22] proposed the ISD algorithm with its subsequent derivaration, for example, MMT [15] and BJMM [3]. Also, Bernstein [4] proposed the quantum version of the Prange algorithm, and Kachigar et al. [13] proposed the quantum version of the

MMT/BJMM algorithms. Perriello et al. [21] introduced the analysis strategy for the previous Hamming CBCs using the Bernstein algorithm. As seen in Section 1.3, we can see some studies about the rank CBC from classical view, but there are few ones from quantum view.

1.5 Our contribution and Organization

In this paper, we propose the quantum version of the GRS algorithm, and compute the computational costs of the quantum GRS algorithm over the quantum circuit. Then, we can find this quantum attack method is effective for RQC by applying to the rank CBCs remained at NIST PQC project 2nd Round. One reason why it is effective is that the value of a certain instance is small. Therefore, we propose the value which RQC is secure against this attack method.

This paper is constructed as following. First, we have already seen the definition of the rank-metric code and the rank SDP. In Section 2, we review the abstract of the classical GRS algorithm. In Section 3, we introduce the quantum computation and the Grover's algorithm [10]. In Section 4, we propose the quantum GRS algorithm based on the above preparations. In Section 5, we consider the attack strategy using the quantum GRS algorithm and the results for ROLLO and RQC. In Section 6, we conclude this paper.

2 Classical GRS algorithm [7]

Suppose that the rank SDP's input q, m, n, k, w, H and s are as in Definition 1. Let r be a positive integer, and let $V_{m,r}$ be an r-dimension entire subspace over $\mathbb{F}_{q^m}$. For $v_m \in V_{m,r}$, let b_m be one of the basis which generates v_m, and $B_{m,r}$ be such the whole b_m, that is, $B_{m,r} = \{b_m \mid v_m \in V_{m,r}\}$. Also, let $V_{n,r}$ be an r-dimension entire subspace over $\mathbb{F}_q^n$, and for $v_n \in V_{n,r}$, let b_n be one of the basis which generates v_n. Let $B_{n,r} = \{b_n \mid v_n \in V_{n,r}\}$. Furthermore, let $\beta = (\beta_1, \cdots, \beta_m)$ be one of the basis over $\mathbb{F}_{q^m}$. For $x \in \mathbb{F}_{q^m}$ and $i \in [1, m]$, there exists $x_i \in \mathbb{F}_q$ such that $x = \sum_{i=1}^{m} x_i \beta_i$. Then, let $p_i : \mathbb{F}_{q^m} \ni x \mapsto x_i \in \mathbb{F}_q$. In the following, we divide the case depending upon which is greater, m or n.

In the case $n \geq m$, we randomly choose r from the closed interval $[w, m - \lceil km/n \rceil]$, and randomly take $F = (F_1, \cdots, F_r)$ from $(B_{m,r})^r$ without duplication. Then, we assume that, for $\ell' \in [1, n]$ and $j \in [1, r]$, there exists $\lambda_{\ell'} = (\lambda_{\ell',1}, \cdots, \lambda_{\ell',r}) \in \mathbb{F}_q^r$ such that $e_{\ell'} = \sum_{j=1}^{r} \lambda_{\ell',j} F_j$, where $e_{\ell'}$ represents the ℓ'-th el-

Algorithm 1 Classical GRS algorithm $(n \geq m)$

```
Input: $q, m, n, k, w, H, s, \beta = (\beta_1, \cdots, \beta_m), r, B_{m,r}$
Output: $e$
 1: $e \leftarrow 0^n$
 2: if $n \geq m$ then
 3:   while $\mathrm{rank}_{\mathbb{F}_q}(e)! = w$ do
 4:     $F = (F_1, \cdots, F_r) \xleftarrow{\$} (B_{m,r})^r$
 5:     for $\ell := 1$ to $n - k$ do
 6:       for $\ell' := 1$ to $n$ do
 7:         $\hat{H}[\ell][\ell'] \leftarrow ()$    {// $\hat{H}[\ell][\ell'] : (m \times r)$ matrix}
 8:         for $j := 1$ to $r$ do
 9:           $H_{\ell,\ell',j} \leftarrow H_{\ell',\ell} \cdot F_r$
10:           $\hat{H}[\ell][\ell'] \leftarrow \hat{H}[\ell][\ell'] \cup (p_1(H_{\ell,\ell',j}), \cdots, p_m(H_{\ell,\ell',j}))^T$
11:       $\hat{s}[\ell] \leftarrow (p_1(s_\ell), \cdots, p_m(s_\ell))$
12:     $(\lambda_{\ell',j})_{1 \leq \ell' \leq n, 1 \leq j \leq r} \leftarrow \mathsf{GE}(\hat{H}, \hat{s})$
13:     for $\ell' := 1$ to $n$ do
14:       $e_{\ell'} \leftarrow \sum_{j=1}^{r} \lambda_{\ell',j} F_j$
15: return $e$
```

ement of $e = (e_1, \cdots, e_n)$. We define $\hat{H} \in \mathbb{F}_q^{m(n-k) \times nr}$ as following ($\ell \in [1, n-k]$):

$$\hat{H} = \begin{pmatrix} \hat{H}[1][1] & \cdots & \hat{H}[1][n] \\ \vdots & \ddots & \vdots \\ \hat{H}[n-k][1] & \cdots & \hat{H}[n-k][n] \end{pmatrix},$$

$$\text{where } \hat{H}[\ell][\ell'] = \begin{pmatrix} p_1(H_{\ell,\ell'}F_1) & \cdots & p_1(H_{\ell,\ell'}F_r) \\ \vdots & \ddots & \vdots \\ p_m(H_{\ell,\ell'}F_1) & \cdots & p_m(H_{\ell,\ell'}F_r) \end{pmatrix}.$$

Also, let $\hat{s}[\ell] = (p_1(s_\ell), \cdots, p_m(s_\ell)) \in \mathbb{F}_q^m$, and we define $\hat{s} = (\hat{s}[1], \cdots, \hat{s}[n-k])$. Then, there exists $\lambda \in \mathbb{F}_q^{nr}$ such that $\hat{H}\lambda = s_p$, where $\lambda[\ell'] = (\lambda_{\ell',1}, \cdots, \lambda_{\ell',r})$ and $\lambda = (\lambda[1], \cdots, \lambda[n])$. $\hat{H}\lambda = s_p$ is equivalent to $He = s$ in the rank SDP. It is neccesary that $nr \geq m(n-k)$ holds for the size of $\hat{H}$ to have a solution at least for this simultaneous equation with an unknown λ. We estimate the range of r from this condition. We evaluate e from λ using the Gaussian Elimination over $\mathbb{F}_q$ for $\hat{H}$ and $\hat{s}$. Hence, the classical GRS algorithm if $n \geq m$ is given in **Algorithm 1**. $F = (F_1, \cdots, F_r) \xleftarrow{\$} (B_{m,r})^r$ described in Line 4 refers to randomly choosing r elements from $B_{m,r}$ without duplication. GE describes in Line 12 refers to the subroutine which receives $(\hat{H}, \hat{s})$, executes the Gaussian Elimination for those, and computes the solution for the simultaneous equation.

Also, in the case $n < m$, we randomly take r from $[w, n-k]$ and $F' = (F'_1, \cdots, F'_r)$ from $(B_{n,r})^r$ without duplication. We define $e' = (e'_1, \cdots, e'_m) \in (\mathbb{F}_q^n)^m$ as following. For $\ell' \in [1, n]$, $e_{\ell'} = \sum_{i=1}^{m} p_i(e_{\ell'}) \cdot \beta_i$ holds. Then, for $i \in$

$[1,m]$, let $e'_i = (p_i(e_1), \cdots, p_i(e_n))$. Then, for $j \in [1,r]$, $i \in [1,m]$, we assume that there exists $\lambda'_{i,j} \in \mathbb{F}_q$ such that $e'_i = \sum_{j=1}^{r} \lambda'_{i,j} F'_j$, where e'_i represents the i-th element of $e' = (e'_1, \cdots, e'_m)$. We define $\hat{H}' \in \mathbb{F}_q^{m(n-k)\times mr}$ as following($i \in [1,m], \ell \in [1,n-k]$):

$$\hat{H}' = \begin{pmatrix} \hat{H}'[1][1] & \cdots & \hat{H}'[1][m] \\ \vdots & \ddots & \vdots \\ \hat{H}'[n-k][1] & \cdots & \hat{H}'[n-k][m] \end{pmatrix},$$

where

$$\hat{H}[\ell][i] = \begin{pmatrix} \sum_{i'=1}^{m} p_{i'}(H_{\ell,1})\beta_{i'} & 0 & \cdots & 0 \\ p_2(H_{\ell,1}) & \sum_{i'=2}^{m} p_{i'}(H_{\ell,2})\beta_{i'} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_n(H_{\ell,1}) & p_n(H_{\ell,2}) & \cdots & \sum_{i'=n}^{m} p_{i'}(H_{\ell,n})\beta_{i'} \\ p_{n+1}(H_{\ell,1}) & p_{n+1}(H_{\ell,2}) & \cdots & \sum_{i'=n+1}^{m} p_{i'}(H_{\ell,n})\beta_{i'} \\ \vdots & \vdots & \ddots & \vdots \\ p_m(H_{\ell,1}) & p_m(H_{\ell,2}) & \cdots & p_m(H_{\ell,n})\beta_m \end{pmatrix}.$$

In detail, for $\hat{\imath} \in [1,m]$, let $\hat{H}[\ell][i]_{\hat{\imath}}$ be the $\hat{\imath}$-th row of $\hat{H}[\ell][i]$ and for $\hat{\ell} \in [1,n]$, let $\hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}}$ be the $\hat{\ell}$-th element of $\hat{H}[\ell][i]_{\hat{\imath}}$. Then, $\hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}}$ is represented as following:

$$\hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}} = \begin{cases} p_{\hat{\imath}}(H_{\ell,\hat{\ell}}) & (1 \leq \hat{\ell} \leq \hat{\imath}-1) \\ \sum_{i'=\hat{\imath}}^{m} p_{i'}(H_{\ell,\min\{\hat{\imath},n\}}) \cdot \beta_{i'} & (\hat{\ell} = \hat{\imath}) \\ 0 & (\hat{\imath}+1 \leq \hat{\ell} \leq n) \end{cases}$$

Also, let $\hat{s}[\ell] = (p_1(s_\ell), \cdots, p_m(s_\ell)) \in \mathbb{F}_q^m$, and we define $\hat{s} = (\hat{s}[1], \cdots, \hat{s}[n-k])$. Then, there exists $\lambda \in \mathbb{F}_q^{nr}$ such that $\hat{H}'\lambda' = \hat{s}$, where $\lambda'[i] = (\lambda'_{i,1}, \cdots, \lambda'_{i,r})$ and $\lambda' = (\lambda'[1], \cdots, \lambda'[m])$. $\hat{H}'\lambda' = \hat{s}$ is equivalent to $He = s$. It is neccesary that $mr \geq m(n-k)$, that is $r \geq n-k$, holds for the size of $\hat{H}'$ to have a solution at least for this simultaneous equation with an unknown λ'. We estimate the range of r from this condition. We evaluate e from λ' using the Gaussian Elimination over $\mathbb{F}_q$ for $\hat{H}'$ and $\hat{s}'$. Therefore, the classical GRS algorithm if $n < m$ is given in **Algorithm 2**.

Let q and x be positive integers, we define $[x]_q := 1 + q + \cdots + q^{x-1}$ and $[x]_q! = \prod_{k=1}^{x} [k]_q$. And let y be a positive integer, and $\binom{x}{y}_q = \frac{[x]_q!}{[x-y]_q![y]_q!}$. Then, let $\ell_{\mathrm{CGRS,n\geq m}}$ be the expected number of loop times in while sentence in in **Algorithm 1**, and $\ell_{\mathrm{CGRS,n\geq m}}$ is given in $\ell_{\mathrm{CGRS,n\geq m}} = \binom{m}{w}_q / \binom{r}{w}_q$ [1]. Also, let $\ell_{\mathrm{CGRS,n<m}}$ be the expected number of iteration times in while sentence in **Algorithm 2**, and $\ell_{\mathrm{CGRS,n<m}}$ is given in $\ell_{\mathrm{CGRS,n<m}} = \binom{n}{w}_q / \binom{r}{w}_q$ [1]. Moreover,

Algorithm 2 Classical GRS algorithm $(n < m)$

```
Input: $q, m, n, k, w, H, s, \beta = (\beta_1, \cdots, \beta_m), r, B_{n,r}$
Output: $e$
1: $e \leftarrow 0^n$
2: if $n \geq m$ then
3:   while $\mathrm{rank}_{\mathbb{F}_q}(e)! = w$ do
4:     $F' = (F'_1, \cdots, F'_r) \xleftarrow{\$} (B_{n,r})^r$
5:     for $\ell := 1$ to $n - k$ do
6:       for $\ell' := 1$ to $n$ do
7:         for $\hat{\ell} := 1$ to $n$ do
8:           for $\hat{\imath} := 1$ to $m$ do
9:             $\hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}} \leftarrow 0$
10:            if $\hat{\ell} == \hat{\imath}$ then
11:              for $i' := \hat{\imath}$ to $m$ do
12:                $\hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}} \leftarrow \hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}} + p_{i'}(H_{\ell,\min\{\hat{\imath},n\}}) \cdot \beta_{i'}$
13:            else if $\hat{\ell} < \hat{\imath}$ then
14:              $\hat{H}[\ell][i]_{\hat{\imath},\hat{\ell}} \leftarrow p_{\hat{\imath}}(H_{\ell,\hat{\ell}})$
15:      $\hat{s}[\ell] \leftarrow (p_1(s_\ell), \cdots, p_m(s_\ell))$
16:    $(\lambda'_{i,j})_{1\leq i\leq m, 1\leq j\leq r} \leftarrow \mathsf{GE}(\hat{H}', \hat{s})$
17:    for $i := 1$ to $m$ do
18:      $e'_i \leftarrow \sum_{j=1}^{r} \lambda'_{i,j} F'_j$
19:      for $\ell' := 1$ to $n$ do
20:        $e_{\ell'} \leftarrow e_{\ell'} + e'_{i,\ell'} \beta_{\ell'}$
21: return $e$
```

the computational complexity of the Gaussian Elimination for the simultaneous equation with $m(n-k)$ unknowns is $O(m^3(n-k)^3)$. So, the computational complexity in the classical GRS algorithm is as following:

$$\begin{cases} O\left((n-k)^3 m^3 \binom{m}{w}_q / \binom{r}{w}_q\right), & n \geq m \\ O\left((n-k)^3 m^3 \binom{n}{w}_q / \binom{r}{w}_q\right), & n < m \end{cases}$$

3 Grover's algorithm

In this section, we explain simply the quantum gates used in this paper and the Grover's algorithm. First, Clifford gate is the set of the quantum gates, i.e., H gate, S gate and CNOT gate, and each of them is represented as follows:

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ \mathrm{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},\ \mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

T gate is the quantum gate represented by $\mathrm{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$, and Clifford+T gate is Clifford gate plus T gate. Z gate is the quantum gate represented by $\mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, and it is written by two S gates as following:

$$\mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \mathrm{SS}$$

We roughly explain about the Grover's algorithm. Let n be a positive integer, V be $\{0,1\}^n$ and M be a non-empty subset of V. And let $f : V \to \{0,1\}$ be the function such that $f(v)$ is 1 if $v \in M$ and 0 otherwise. The Grover's algorithm is the quantum algorithm to search $x_0 \in M$ taking (V, f) as inputs. This algorithm has the computational complexity of $O\left(\sqrt{\frac{|V|}{|M|}}\right)$. Let H^V is the Hilbert space associated with V. U_o and U_d are the unitary operators over H^V and defined as following:

$$U_o(|i\rangle) := \begin{cases} -|i\rangle & i \in M; \\ |i\rangle & \text{o.w.}; \end{cases}$$
$$U_d(|i\rangle) := (2\mathrm{H}^{\otimes n}|0\rangle\langle 0|\mathrm{H}^{\otimes n} - I_n)|i\rangle.$$

$\mathrm{H}^{\otimes n}$ denotes $\underbrace{\mathrm{H} \otimes \cdots \otimes \mathrm{H}}_{n}$, that is, the Tensor products of n H gates. U_o is called the oracle operator and U_d is the unitary operator called diffuser. Then, the Grover's algorithm is written by **Algorithm 3**.

Algorithm 3 Grover's algorithm

```
Input: $V \subset \{0,1\}^n,\ f : V \to \{0,1\}$
Output: $x_0 \in \{0,1\}^n$ s.t. $f(x_0) = 1$
1: $|\psi\rangle \leftarrow |0^n\rangle$
2: $|\psi\rangle \leftarrow H^{\otimes n}|\psi\rangle$
3: for $i := 1$ to $\left\lfloor \frac{\pi}{4\arcsin(\sqrt{\frac{|M|}{|V|}})} \right\rfloor$ do
4:     $|\psi\rangle \leftarrow U_o|\psi\rangle$
5:     $|\psi\rangle \leftarrow U_d|\psi\rangle$
6: return $|\psi\rangle$
```

4 Quantum GRS algorithm

In this section, we propose the quantum GRS algorithm by combining the classical GRS algorithm and the Grover's algorithm. In quantum GRS algorithm, by using the Grover's algorithm, we can efficiently probe a set F of the basis.

Suppose $n \geq m$. Denote, by $\mathsf{Grover}_{n\geq m}$, the subroutine using the Grover's algorithm. Let $B_{m,r}$ be V in the Grover's algorithm. Then, $|B_{m,r}| = \binom{m}{r}_q$ holds. Also, let F_{m-w} be a subspace over $\mathbb{F}_{q^m}$ whose dimension is $m-w$, and V_{m-w} be such a whole F_{m-w}. And let $V_{m-w,r-w}$ be an $(r-w)$-dim entire subspace included in V_{m-w} Then, let $F_{m-w,r-w}$ be one of the subspaces in $V_{m-w,r-w}$, and $b_{m-w,r-w}$ be one of the basis of $F_{m-w,r-w}$. That is, $B_{m-w,r-w} = \{b_{m-w} \mid F_{m-w,r-w} \in V_{m-w,r-w}\}$, and it implies $|B_{m-w,r-w}| = \binom{m-w}{r-w}$. Let M in the Grover's algorithm be $B_{m-w,r-w}$, and we take f as following. For $v \in V$, we construct $\hat{H}$ and $\hat{s}$ from (H, s, β, v) and evaluate $(\lambda_{\ell',j})_{\ell',j}$ as in Line 5 in **Algorithm 1**, and evaluate e as in Line 7. Then, if $\mathrm{rank}_{\mathbb{F}_q}(e) = w$, f returns 1 which indicates $v \in M$, and returns 0 otherwise. Then, $|M|$ is equal to the number of r-dim subspaces including the subspace F_e, generated by e, whose dimension is w. Such a number is equal to the number of $(n-w)$-dimension subspaces in the subspace from $\mathbb{F}_{q^m}^m$ minus F_e, so $|M| = \binom{m-w}{r-w}$. Here,

$$\frac{\binom{m}{r}_q}{\binom{m-w}{r-w}_q} = \frac{\frac{[m]_q!}{[m-r]_q![r]_q!}}{\frac{[m-w]_q!}{[m-r]_q![r-w]_q!}} = \frac{\frac{[m]_q!}{[r]_q!}}{\frac{[m-w]_q!}{[r-w]_q!}} = \frac{\frac{[m]_q!}{[m-w]_q!}}{\frac{[r]_q!}{[r-w]_q!}} = \frac{\frac{[m]_q!}{[m-w]_q![w]_q!}}{\frac{[r]_q!}{[r-w]_q![w]_q!}} = \frac{\binom{m}{w}_q}{\binom{r}{w}_q}.$$

Hence, by $|V|/|M| = \binom{m}{w}_q/\binom{r}{w}_q$, $|V|/|M|$ coincides with the number of iteration in while sentence in **Algorithm 1**. That is, in $\mathsf{Grover}_{n\geq m}$, the operations corresponding to the while sentence in **Algorithm 1** are executed in $\ell_{\mathrm{QGRS},n\geq m}(= \sqrt{\ell_{\mathrm{CGRS},n\geq m}})$ times. Thus, $\mathsf{Grover}_{n\geq m}$ can be constructed with V, M and f shown above, and given by **Algorithm 4**.

QRA appearing in Line 6 refers to the quantum random access [11]. Let $A = (a_1, \cdots, a_n)$ be an n element array each of whose element is m-bits. For A, $1 \leq i \leq m$ and $b \in \mathbb{F}_2^m$, we think about the quantum circuit to compute $b \oplus' a_i$, where $\oplus'$ means the addition in $\mathbb{F}_2^m$. That is, the quantum circuit which returns

Algorithm 4 $\mathsf{Grover}_{n\geq m}$

```
Input: $w, H, s, \beta, B_{m,r}$
Output: $F$
 1: $|\psi\rangle \leftarrow |0^{|B_{m,r}|}\rangle$
 2: $|\psi\rangle \leftarrow H^{\otimes|B_{m,r}|}|\psi\rangle$
 3: for $x := 1$ to $\sqrt{\frac{\binom{m}{w}_q}{\binom{r}{w}_q}}$ do
 4:   $|e\rangle = |e_1\rangle\cdots|e_r\rangle \leftarrow |0^{\log q}\rangle\cdots|0^{\log q}\rangle$
 5:   for $j := 1$ to $r$ do
 6:     $|F_{\psi,j}\rangle \leftarrow \mathrm{QRA}((\mathrm{QRA}(|B_{m,r}\rangle, |\psi\rangle)), |j\rangle)$
 7:   for $\ell := 1$ to $n-k$ do
 8:     for $\ell' := 1$ to $n$ do
 9:       for $j := 1$ to $r$ do
10:         $|\hat{H}_{\ell,\ell',j}\rangle \leftarrow |p_1(|H_{\ell',\ell}\rangle \cdot |F_{\psi,j}\rangle)\rangle\cdots|p_m(|H_{\ell',\ell}\rangle \cdot |F_{\psi,j}\rangle)\rangle$
11:   for $\ell := 1$ to $n-k$ do
12:     $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle\cdots|p_m(|s_\ell\rangle)\rangle$
13:   $(|\lambda_{\ell',j}\rangle)_{1\leq\ell'\leq n, 1\leq j\leq r} \leftarrow \mathsf{GE}(|\hat{H}\rangle, |\hat{s}\rangle)$
14:   for $\ell' := 1$ to $n$ do
15:     $|e_{\ell'}\rangle \leftarrow \sum_{j=1}^{r} |\lambda_{\ell',j}\rangle \cdot |F_{\psi,j}\rangle$
16:   if $\mathrm{rank}_{\mathbb{F}_q}(|e\rangle)! = w$ then
17:     $|\psi\rangle \leftarrow -|\psi\rangle$
18:   $|\psi\rangle \leftarrow U_d|\psi\rangle$
19: return $\mathrm{QRA}(|B_{m,r}\rangle, |\psi\rangle)$
```

Algorithm 5 Quantum GRS algorithm $(n \geq m)$

```
Input: $q, m, n, k, w, H, s, \beta = (\beta_1, \cdots, \beta_m), r, B_{m,r}$
Output: $|e\rangle$
 1: $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
 2: $|F\rangle = |F_1\rangle \cdots |F_r\rangle \leftarrow \mathsf{Grover}_{n \geq m}(w, H, s, \beta, B_{m,r})$
 3: for $\ell := 1$ to $n-k$ do
 4:     for $\ell' := 1$ to $n$ do
 5:         for $j := 1$ to $r$ do
 6:             $|\hat{H}_{\ell,\ell',j}\rangle \leftarrow |p_1(|H_{\ell',\ell}\rangle \cdot |F_j\rangle)\rangle \cdots |p_m(|H_{\ell',\ell}\rangle \cdot |F_j\rangle)\rangle$
 7: for $\ell := 1$ to $n-k$ do
 8:     $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
 9: $(|\lambda_{\ell',j}\rangle)_{1 \leq \ell' \leq n, 1 \leq j \leq r} \leftarrow \mathsf{GE}(|\hat{H}\rangle, |\hat{s}\rangle)$
10: for $\ell' := 1$ to $n$ do
11:     for $j := 1$ to $r$ do
12:         $|e_{\ell'}\rangle \leftarrow |e_{\ell'}\rangle + |\lambda_{\ell',j}\rangle \cdot |F_j\rangle$
13: return $|e\rangle$
```

$(|i\rangle, |b \oplus' a_i\rangle, |A\rangle)$ for an input $(|i\rangle, |b\rangle, |A\rangle)$. Such a quantum circuit is called the quantum random access. Especially, when $b = 0^m$, we define $\mathrm{QRA}(|A\rangle, |i\rangle) = |a_i\rangle$ using $(|i\rangle, |0^m\rangle, |A\rangle) \mapsto (|i\rangle, |a_i\rangle, |A\rangle)$. Also, $|p_1(|s_\ell\rangle)\rangle$ in Line 10 represents the quantum state corresponding to $p_1(s_\ell)$. Moreover, GE in Line 13 refers to the subroutine which obtains the solution $\lambda_{\ell',j}$ for the simultaneous equation derived from $|H\rangle$ and $|s\rangle$ applying the Gaussian Elimination. The quantum GRS algorithm in the case $n \geq m$ is given in **Algorithm 5** by using this subroutine.

Similarly, we can construct the subroutine $\mathsf{Grover}_{n<m}$ for $n < m$. That is, let V and M in the Grover's algorithm be $B_{n,r}$ and $B_{n-w,r-w}$, respectively, and we have only to define f to be the same with that for $n \geq m$. Therefore, the number of iterations, $\mathsf{Grover}_{n>m}$, in **Algorithm 6** is $\ell_{\mathrm{QGRS},n<m} = \sqrt{\ell_{\mathrm{CGRS},n<m}}$. Thus, we construct V, M and f, and the algorithm for $\mathsf{Grover}_{n<m}$ is given in **Algorithm 6**. Therefore, the quantum GRS algorithm when $n < m$ is written in **Algorithm 7** based on the above subroutine.

The arguments in the quantum GRS algorithm are the same as those in the classical one. Hence, the computational complexity in the quantum GRS algorithm is as following:

$$\begin{cases} O\left((n-k)^3 m^3 \sqrt{\binom{m}{w}_q / \binom{r}{w}_q}\right), & n \geq m; \\ O\left((n-k)^3 m^3 \sqrt{\binom{n}{w}_q / \binom{r}{w}_q}\right), & n < m. \end{cases}$$

In the following, we propose the improved quantum GRS algorithm which can be more efficient by the improvement for the quantum GRS algorithm. When $n \geq m$, let $B'_{m,r}$ be a set of $|B_{m,r}|/|B_{m-w,r-w}|$-element basis randomly selected from $B_{m,r}$ without duplication. Let V in the Grover's algorithm be $B'_{m,r}$. Define M and f to be the same in **Algorithm 5**, and we can get the expected value of $|M|$ is 1. Define $\mathsf{Improved\_Grover}_{n \geq m}$ as the above subroutine, and we can describe $\mathsf{Improved\_Grover}_{n \geq m}$ as in **Algorithm 8**.

Algorithm 6 Grover$_{n<m}$

```
Input: $w, H, s, \beta, B_{n,r}$
Output: $F'$
1: $|\psi\rangle \leftarrow |0^{|B_{n,r}|}\rangle$
2: $|\psi\rangle \leftarrow H^{\otimes |B_{n,r}|}|\psi\rangle$
3: for $x := 1$ to $\sqrt{\frac{\binom{n}{w}_q}{\binom{r}{w}_q}}$ do
4:     $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
5:     for $j := 1$ to $r$ do
6:         $|F'_{\psi,j}\rangle \leftarrow \mathrm{QRA}((\mathrm{QRA}(|B_{n,r}\rangle, |\psi\rangle)), |j\rangle)$
7:     for $\ell := 1$ to $n-k$ do
8:         for $i := 1$ to $m$ do
9:             for $i' := 1$ to $m$ do
10:                for $j := 1$ to $r$ do
11:                    $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |0\rangle^{\lceil \log_2 q \rceil}$
12:                    for $\ell' := 1$ to $n$ do
13:                        if $\hat{i} <= n$ then
14:                            $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,\hat{i}}\rangle) \cdot |\beta_{i'}\rangle$
15:                        else
16:                            $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,n}\rangle) \cdot |\beta_{i'}\rangle$
17:    for $\ell := 1$ to $n-k$ do
18:        $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
19:    $(|\lambda_{\ell',j}\rangle)_{1 \le \ell' \le n, 1 \le j \le r} \leftarrow \mathsf{GE}(|\hat{H}'\rangle, |\hat{s}\rangle)$
20:    for $i := 1$ to $m$ do
21:        $|e'_i\rangle \leftarrow |0^{\log q}\rangle$
22:        for $j := 1$ to $r$ do
23:            $|e'_i\rangle \leftarrow |e'_i\rangle + |\lambda'_{\ell',j}\rangle \cdot |F'_{\psi,j}\rangle$
24:        for $\ell' := 1$ to $n$ do
25:            $|e_{\ell'}\rangle \leftarrow |e_{\ell'}\rangle + |e'_{i,\ell'}\rangle|\beta_{\ell'}\rangle$
26:    if $\mathrm{rank}_{\mathbb{F}_q}(|e\rangle)! = w$ then
27:        $|\psi\rangle \leftarrow -|\psi\rangle$
28:    $|\psi\rangle \leftarrow U_d|\psi\rangle$
29: return $\mathrm{QRA}(|B_{n,r}\rangle, |\psi\rangle)$
```

Algorithm 7 Quantum GRS algorithm $(n < m)$

```
Input: $q, m, n, k, w, H, s, \beta = (\beta_1, \cdots, \beta_m), r, B_{n,r}$
Output: $e$
 1: $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
 2: $|F'\rangle = |F'_1\rangle \cdots |F'_r\rangle \leftarrow \mathsf{Grover}_{n<m}(w, H, s, \beta, B_{n,r})$
 3: for $\ell := 1$ to $n - k$ do
 4:   for $i := 1$ to $m$ do
 5:     for $i' := 1$ to $m$ do
 6:       for $j := 1$ to $r$ do
 7:         $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |0\rangle^{\lceil \log_2 q \rceil}$
 8:         for $\ell' := 1$ to $n$ do
 9:           if $\hat{i} <= n$ then
10:             $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,\hat{i}}\rangle) \cdot |\beta_{i'}\rangle$
11:           else
12:             $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,n}\rangle) \cdot |\beta_{i'}\rangle$
13: for $\ell := 1$ to $n - k$ do
14:   $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
15: $(|\lambda_{\ell',j}\rangle)_{1 \le \ell' \le n, 1 \le j \le r} \leftarrow \mathsf{GE}(|\hat{H}'\rangle, |\hat{s}\rangle)$
16: for $i := 1$ to $m$ do
17:   $|e'_i\rangle \leftarrow |0^{\log q}\rangle$
18:   for $j := 1$ to $r$ do
19:     $|e'_i\rangle \leftarrow |e'_i\rangle + |\lambda'_{\ell',j}\rangle \cdot |F'_j\rangle$
20:   for $\ell' := 1$ to $n$ do
21:     $|e_{\ell'}\rangle \leftarrow |e_{\ell'}\rangle + |e'_{i,\ell'}\rangle |\beta_{\ell'}\rangle$
22: return $|e\rangle$
```

Algorithm 8 Improved_Grover$_{n\geq m}$

```
Input: $w, H, s, \beta, B'_{m,r}$
Output: $F$
 1: $|\psi\rangle \leftarrow |0^{|B_{m,r}|'}\rangle$
 2: $|\psi\rangle \leftarrow H^{\otimes|B_{m,r}|'}|\psi\rangle$
 3: for $x := 1$ to $\sqrt{\frac{\binom{m}{w}_q}{\binom{r}{w}_q}}$ do
 4:     $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
 5:     for $j := 1$ to $r$ do
 6:         $|F_{\psi,j}\rangle \leftarrow \mathrm{QRA}((\mathrm{QRA}(|B'_{m,r}\rangle, |\psi\rangle)), |j\rangle)$
 7:     for $\ell := 1$ to $n-k$ do
 8:         for $\ell' := 1$ to $n$ do
 9:             for $j := 1$ to $r$ do
10:                 $|\hat{H}_{\ell,\ell',j}\rangle \leftarrow |p_1(|H_{\ell',\ell}\rangle \cdot |F_{\psi,j}\rangle)\rangle \cdots |p_m(|H_{\ell',\ell}\rangle \cdot |F_{\psi,j}\rangle)\rangle$
11:     for $\ell := 1$ to $n-k$ do
12:         $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
13:     $(|\lambda_{\ell',j}\rangle)_{1\leq\ell'\leq n, 1\leq j\leq r} \leftarrow \mathsf{GE}(|\hat{H}\rangle, |\hat{s}\rangle)$
14:     for $\ell' := 1$ to $n$ do
15:         $|e_{\ell'}\rangle \leftarrow \sum_{j=1}^{r} |\lambda_{\ell',j}\rangle \cdot |F_{\psi,j}\rangle$
16:     if $\mathrm{rank}_{\mathbb{F}_q}(|e\rangle)! = w$ then
17:         $|\psi\rangle \leftarrow -|\psi\rangle$
18:     $|\psi\rangle \leftarrow U_d|\psi\rangle$
19: return $\mathrm{QRA}(|B_{m,r}\rangle, |\psi\rangle)$
```

Algorithm 9 Improved quantum GRS algorithm $(n \geq m)$

```
Input: $q, m, n, k, w, H, s, \beta = (\beta_1, \cdots, \beta_m), r, B'_{m,r}$
Output: $e$
 1: $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
 2: $|F\rangle = |F_1\rangle \cdots |F_r\rangle \leftarrow \mathsf{Improved\_Grover}_{n \geq m}(w, H, s, \beta, B'_{m,r})$
 3: for $\ell := 1$ to $n-k$ do
 4:     for $\ell' := 1$ to $n$ do
 5:         for $j := 1$ to $r$ do
 6:             $|\hat{H}_{\ell,\ell',j}\rangle \leftarrow |p_1(|H_{\ell',\ell}\rangle \cdot |F_j\rangle)\rangle \cdots |p_m(|H_{\ell',\ell}\rangle \cdot |F_j\rangle)\rangle$
 7: for $\ell := 1$ to $n-k$ do
 8:     $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
 9: $(|\lambda_{\ell',j}\rangle)_{1 \leq \ell' \leq n, 1 \leq j \leq r} \leftarrow \mathsf{GE}(|\hat{H}\rangle, |\hat{s}\rangle)$
10: for $\ell' := 1$ to $n$ do
11:     for $j := 1$ to $r$ do
12:         $|e_{\ell'}\rangle \leftarrow |e_{\ell'}\rangle + |\lambda_{\ell',j}\rangle \cdot |F_j\rangle$
13: return $|e\rangle$
```

The improved quantum GRS algorithm is given in **Algorithm 9** by using this subroutine.

Similarly for $n < m$, let $B'_{n,r}$ be a $|B_{n,r}|/|B_{n-w,r-w}|$ element subset taking randomly from $B_{n,r}$. The improved quantum GRS algorithm in the case $n < m$ is given in **APPENDIX B.2**. The asymptotic computational complexity of the improved quantum GRS algorithm is equal to that of the quantum GRS algorithm. The improved quantum GRS algorithm has the advantage that the number of H gate used in the superposition can be reduced because the size of V in the Grover's algorithm can be smaller.

5 Analysis

We discuss the attack method using the quantum GRS algorithm and its result for the rank SDP with the parameters given in **Table 1**. Suppose $q = 2$ and β be the standard basis over $\mathbb{F}_{q^m}$, that is, $\beta = (\beta_0, \beta_1, \cdots, \beta_{m-1}) = (1, 2, \cdots, 2^{m-1})$. Also, **Table 1** shows the rank SDP instances which correspond to the ROLLO and RQC, the cryptosystems remained at the 2nd Round of the NIST PQC standardization project, with the security levels of 128, 192 and 256.

In this section, we introduce the G-cost, D-cost and W-cost to estimate the computational costs for the quantum GRS algorithm and the improved quantum GRS algorithm. Then, we think of quantum circuits that execute the various operations in those quantum algorithms. We can combine these circuits to form quantum circuits that execute quantum GRS algorithms. Therefore, we can evaluate the computational costs for the quantum GRS algorithm and the improved version of that algorithm by finding the various operations used in the quantum GRS algorithms and the number of times each is executed. Also, we can compute the computational costs of the attack algorithms, that is quantum GRS

cryptosystem	security bit	m	n	k	w
ROLLO	128	67	166	83	7
	192	79	194	97	8
	256	97	226	113	9
RQC	128	127	113	3	7
	192	151	149	5	8
	256	181	179	3	9

Table 1. targeted cryptosystem and security bit

algorithm and improved quantum GRS algorithm, for each cryptosystem by using the instances (m, n, k, w) in **Table 1**. Hence, we can determine whether the cryptosystem in **Table 1** is secure or not, by comparing the computational costs of the attack algorithms for each cryptosystem with each security level with the computational costs for the circuit corresponding to that security bit.

5.1 Introducing the new computational costs

We introduce the computational costs explained in [11]. Let C be a quantum circuit that consists of Clifford+T gates. The G-cost means the number of all the quantum gates appearing in C. The D-cost represents the depth of C, and the W-cost refers to the number of qubits in C. These computational costs are evaluated by $\log_2$. In this paper, we mainly compare the G-cost for a circuit corresponding to the security bit with the G-cost obtained by our strategy from the parameters (m, n, k, w) in **Table 1**. We can estimate the computational costs (G, D and W) of the quantum GRS algorithms that simulate over classical circuits or quantum circuits. The latter ones are the computational costs of quantum circuits which are consisting of the Clifford+T gate, and which execute the quantum GRS algorithms. The former ones are the computational costs restricted to the classical PRAM operations for the latter quantum circuit. As seen in [11], also in this paper, in the case estimating the computational costs of the quantum GRS algorithms over classical circuits, we do not think about a superposition of the quantum states. This is because a superposition which can be executed in quantum RAM operations cannot be simulated in classical ones. Therefore, we do not take account of the computational costs of the Grover's algorithm itself. Then, in the improved quantum GRS algorithm, since we alter only the part concerned with the quantum RAM operations, the computational costs of the classical circuit for the quantum GRS algorithm remain the same with those of the classical circuit for the improved version of that algorithm. We consider the G-cost, D-cost and W-cost about the operations in the quantum GRS algorithms. Also, we think the numbers of the input qubits and the ancilla bits to estimate the W-cost. In the following, let m and n be positive integers, and let $a, b \in \mathbb{F}_{2^m}$ and $e \in \mathbb{F}_{2^m}^n$. Also, let β be the standard basis over $\mathbb{F}_{2^m}$.

Algorithm 10 3 adder for classical bits

```
Input: $a, b, c$
Output: $s, d$
1: $s, d \leftarrow 0$
2: if $a \odot b = 1$ then
3:     $s \leftarrow s \oplus 1$
4: $d \leftarrow a \oplus b$
5: if $c \odot d = 1$ then
6:     $s \leftarrow s \oplus 1$
7: $d \leftarrow c \oplus d$
8: return $s, d$
```

Here, we introduce the computational costs for the quantum circuits that execute the Gaussian Elimination and the 3 adder. 3 adder is an operation that, for $a, b, c \in \mathbb{F}_2$, evaluates $s, d \in \mathbb{F}_2$ such that $|a \oplus b \oplus c\rangle = |s\rangle|d\rangle$ with $a+b+c = 2s+d \in \mathbb{N}$, where $\oplus$ means the addition in $\mathbb{F}_2$. In the following, let x and y be positive integers, for a matrix $A \in \mathbb{F}_2^{x \times y}$, let $|A\rangle$ be a quantum state corresponding to A. For $\hat{H} \in \mathbb{F}_2^{x \times y}$, let $U \in \mathbb{F}_2^{x \times x}$ and $Q = U\hat{H} \in \mathbb{F}_2^{x \times y}$, where U is a matrix to execute the Gaussian Elimination for $\hat{H}$. Then, we consider the quantum circuit to take $|\hat{H}\rangle$ and return $|U\rangle$ and $|Q\rangle$. Perriello et al. [21] introduced the quantum circuit which executes the Gaussian Elimination, and such a quantum circuit has xy input qubits, $\frac{3}{2}(x-1)x$ ancilla qubits, the G-cost of $(x-1)\{x(36y - 20x + \frac{43}{2}) + 8\}$ and the D-cost of $\frac{8}{3}(x-1)x(9y - 4x + 5)$.

Let $a, b, c, s, d \in \mathbb{F}_2$, in a classical circuit, the 3 adder is given in **Algorithm 10**, where $\odot$ means the multiplication in $\mathbb{F}_2$. The quantum circuit which executes $\odot$ for two quantum states can be realized with one Toffoli gate. Here, Toffoli gate is the quantum gate expressed as following:

$$\text{Toffoli} = \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0 \end{pmatrix}.$$

The construction for Toffoli gate by Clifford+T gates is given in [24] by Shende. That quantum circuit which executes the Toffoli gate has 3 input qubits, no ancilla bit, the G-cost of 24 and the D-cost of 16. Then, that quantum circuit which executes the 3 adder is given in **Fig. 1** and has 3 input qubits, 2 ancilla bits, the G-cost of 51 and the D-cost of 32.

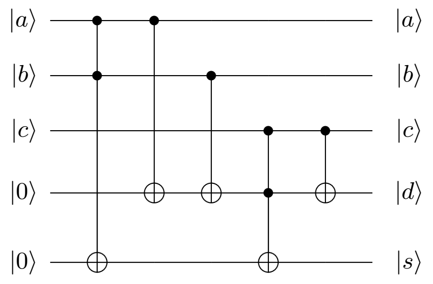

Fig. 1. 3 adder

Algorithm 11 the addition for classical bits

```
Input: $a = a_1 \cdots a_m, b = b_1 \cdots b_m$
Output: $c = c_1 \cdots c_m$
1: for $i := m$ downto 1 do
2:     if $i == m$ then
3:         $s_m, c_m \leftarrow$3-adder$(0, a_m, b_m)$
4:     else
5:         $s_i, c_i \leftarrow$3-adder$(s_{i+1}, a_i, b_i)$
6: return $c$
```

5.2 The computational costs for the addition of quantum bits over $\mathbb{F}_{2^m}$

For $a, b \in \mathbb{F}_{2^m}$, we define $|a + b\rangle$ as the addition for a and b, where $+$ means the addition over $\mathbb{F}_{2^m}$. In the following, we expand a and b with β, and consider $a = a_1 \cdots a_m, b = b_1 \cdots b_m \in \mathbb{F}_{2^m}$. In a classical circuit, the algorithm that takes a and b and outputs $c = c_1 \cdots c_m$ such that $c = a + b \bmod 2^m$ is described in **Algorithm 11**. **Fig. 2** shows the quantum circuit which outputs $c_1, c_2 \in \mathbb{F}_2$ such that $|a_1a_2\rangle + |b_1b_2\rangle = |a_1a_2 + b_1b_2\rangle = |c_1c_2\rangle$ for inputs $|a\rangle = |a_1a_2\rangle$ and $|b\rangle = |b_1b_2\rangle$ using three ancilla bits including $|c_1\rangle$ and $|c_2\rangle$. **Fig. 2** can be constructed from a Toffoli gate and 5 CNOT gates. Therefore, for $a, b \in \mathbb{F}_{2^m}$, the quantum circuit which executes $|a\rangle + |b\rangle$ can be realized with $(m - 1)$ Toffoli gates and $(2m + 1)$ CNOT gates. Hence, such a quantum circuit has $2m$ input qubits, $(2m - 1)$ ancilla bits, the G-cost of $26m - 23$ and the D-cost of 16, respectively.

5.3 The computational costs for the product of quantum bits over $\mathbb{F}_{2^m}$

For $a, b \in \mathbb{F}_{2^m}$, we define $|a \cdot b\rangle$ as the product for a and b, where $\cdot$ means the multiplication over $\mathbb{F}_{2^m}$. As in Section 5.2, let $a = a_1 \cdots a_m, b = b_1 \cdots b_m$, and $a = a_1 \cdot 2^{m-1} + \cdots a_{m-1} \cdot 2^1 + a_m \cdot 2^0$ holds in $\mathbb{F}_{2^m}$. For $1 \le i \le m$, we think to compute each $c_i \in \mathbb{F}_2$ such that $a \cdot b = c_1 \cdot 2^{m-1} + \cdots + c_{m-1}2^1 + c_m 2^0 \bmod 2^m$. In a classical circuit, such a multiplier can be realized with Full-Adders, and that algorithm is written in **Algorithm 12**. For example, if $m = 3$, the variables are

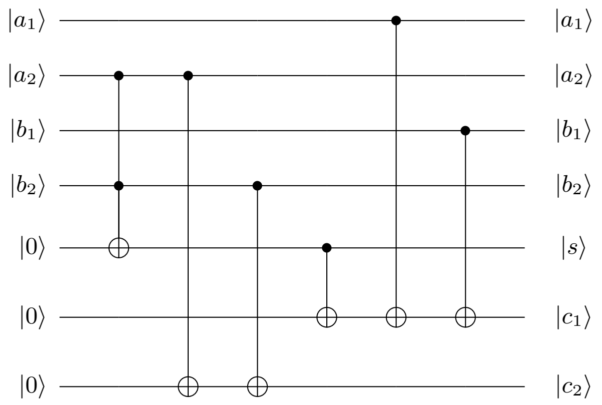

Fig. 2. The addition for qubits

determined as following:

$$\begin{aligned} c_3 &\leftarrow a_3 \odot b_3 \\ c_{5,1}, s_{5,1} &\leftarrow \text{3-adder}(0, a_2 \odot b_3, a_3 \odot b_2) \\ c_2 &\leftarrow c_{5,1} \\ c_{4,1}, s_{4,1} &\leftarrow \text{3-adder}(s_{5,1}, a_1 \odot b_3, a_2 \odot b_2) \\ c_{4,2}, s_{4,2} &\leftarrow \text{3-adder}(0, c_{4,1}, a_3 \odot b_1) \\ c_1 &\leftarrow c_{4,2} \end{aligned}$$

Fig. 3 shows the quantum circuit which outputs c_1, c_2 and $c_3 \in \mathbb{F}_2$ such that $|a_1a_2a_3\rangle \cdot |b_1b_2b_3\rangle = |a_1a_2a_3 \cdot b_1b_2b_3\rangle = |c_1c_2c_3\rangle$ for the input of $|a\rangle = |a_1a_2a_3\rangle$ and $|b\rangle = |b_1b_2b_3\rangle$. In **Fig. 3**, the qubits $|s_{23,32}\rangle, |s_{31,22}\rangle$ and $|s_1\rangle$ denote the carry of digit in the 3 adder. Also, the qubit $|c_{31,22}\rangle$ satisfies $|s_{23,32}\rangle + |a_3 \cdot b_1\rangle + |a_2 \cdot b_2\rangle = |s_{31,22}c_{31,22}\rangle$. In addition, in each rectangle representing the 3 adder in **Fig. 3**, three '$\rightarrow$' crossing the left vertical edge and two '$\rightarrow$' crossing the right one express the inputs and the outputs for the 3 adder, respectively. For example, the left-most 3 adder can realize the operation $|0\rangle + |a_3 \cdot b_2\rangle + |a_2 \cdot b_3\rangle = |s_{23,32}c_2\rangle$. **Fig. 3** can be realized with 6 Toffoli gates and 3 '3 adders'. Therefore, in general, for $a, b \in \mathbb{F}_{2^m}$, the quantum circuit which executes $|a\rangle \cdot |b\rangle$ can be constructed from $m(m+1)/2$ Toffoli gates and $(m-1)/2$ '3 adders'. That quantum circuit has $2m$ input qubits, $m^2 + m + 1$ ancilla bits, the G-cost of $9m(25m-9)/2$ and the D-cost of $16m$.

5.4 The computational costs for estimating $\mathrm{rank}_{\mathbb{F}_2}(e)$

For $e = (e_1, \cdots, e_n) \in \mathbb{F}_{2^m}^n$ and $1 \leq \ell' \leq n$, by expanding $e_{\ell'}$ with β, we see $e_{\ell'}$ as $e_{\ell'} = (e_{\ell',1}, \cdots, e_{\ell',m}) \in \mathbb{F}_2^m$ and e as $e \in \mathbb{F}_2^{n \times m}$. By the definition of $\mathrm{rank}_{\mathbb{F}_2}(e)$, $\mathrm{rank}_{\mathbb{F}_2}(e)$ is $\dim(\mathrm{Im}(\varphi_e))$, and $\dim(\mathrm{Im}(\varphi_e))$ for $e \in \mathbb{F}_{2^m}^n$ coincides with $\mathrm{rank}(e)$, the rank of the matrix $e \in \mathbb{F}_2^{n \times m}$. From the above, for $e \in \mathbb{F}_{2^m}^n$, to compute

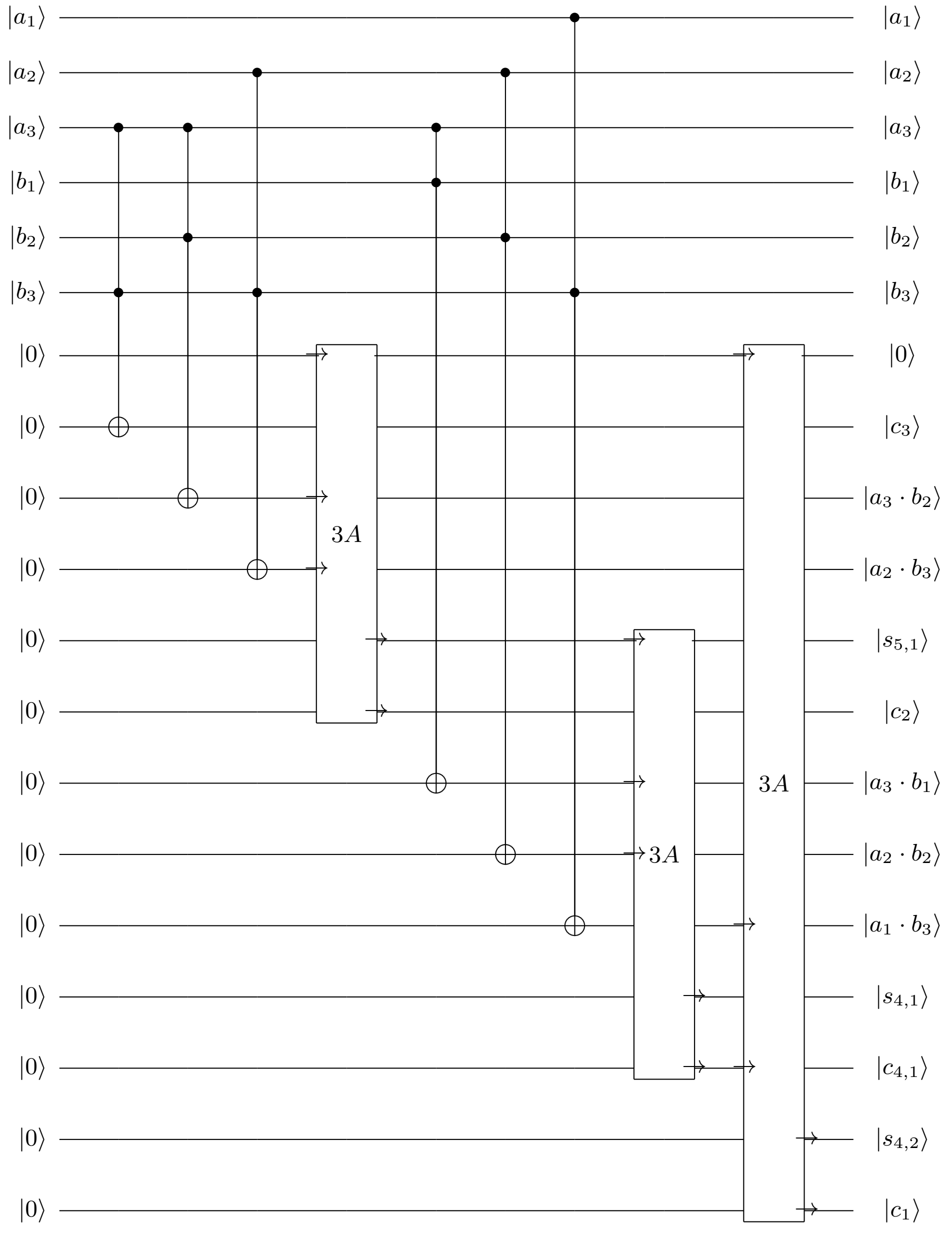

Fig. 3. The multiplication for qubits

Algorithm 12 the multiplication for classical bits

```
Input: $a = a_1 \cdots a_m, b = b_1 \cdots b_m$
Output: $c = c_1 \cdots c_m$
 1: $c_m \leftarrow a_m \odot b_m$
 2: for $i := m-1$ downto 1 do
 3:     for 3A_$i := 1$ to $m-i$ do
 4:         if 3A_$i == m-1$ then
 5:             $c_{m+i,3\text{A}\_i}, s_{m+i,3\text{A}\_i} \leftarrow$ 3-adder$(0, c_{m+i,3\text{A}\_i-1}, a_{m-i+1} \odot b_{m-1})$
 6:         else if 3A_$i == 1$ then
 7:             $c_{m+i,3\text{A}\_i}, s_{m+i,3\text{A}\_i} \leftarrow$ 3-adder$(s_{m+i+1,3\text{A}\_i}, a_{m-i} \odot b_m, a_{m-i+1} \odot b_{m-1})$
 8:         else
 9:             $c_{m+i,3\text{A}\_i}, s_{m+i,3\text{A}\_i} \leftarrow$ 3-adder$(s_{m+i+1,3\text{A}\_i}, c_{m+i,3\text{A}\_i-1}, a_{m-i+1} \odot b_{m-1})$
10:     $c_i \leftarrow c_{m+i,m-i}$
11: return $c$
```

$\text{rank}_{\mathbb{F}_2}(e)$, we consider e as the $n \times m$ matrix over $\mathbb{F}_2$, and execute the Gaussian Elimination for $e \in \mathbb{F}_2^{n\times m}$. We need nm ancilla bits and nm CNOT gates as the copy of the input qubits because the value of the input qubits are altered during the execution of the Gaussian Elimination. Therefore, that quantum circuit has nm input qubits, nm ancilla bits, the G-cost of $\text{GE}(n,m) + nm$ and the D-cost of $16(m+1)$.

5.5 The computational costs for quantum random access

For $i \in [1,n]$ and $A = (a_1, \cdots, a_n) \in (\mathbb{F}_2^m)^n$, we think the quantum circuit that executes the quantum random access. By [11], such a quantum circuit has $\log n + m + mn$ input qubits, $nm + n\log n$ ancilla bits, the G-cost of $nm + n\log n$ and the D-cost of $\log m + \log n$.

5.6 The computational costs for a superposition

5.6.1 In the case $n \geq m$

A superposition can be realized with $|B_{m,r}| = \binom{m}{r}_q$-H gates. That is, the quantum circuit which executes the superposition has $|B_{m,r}|$ input qubits, no ancilla bits, the G-cost of $|B_{m,r}|$ and the D-cost of 1.

5.6.2 In the case $n < m$

In this case as well, a superposition can be constructed with $|B_{n,r}| = \binom{n}{r}_q$-H gates. Then, such a quantum circuit has $|B_{n,r}|$ input qubits, no ancilla bits, the G-cost of $|B_{n,r}|$ and the D-cost of 1.

5.7 The computational costs for oracle phase flipping

In the Grover's algorithm, V and M denote the set of all elements and targeted elements, respectively. Then, U_o can be constructed with some X gates. There-

Operation	Abbreviation	Argument	Denotion
Gaussian Elimination	GE	(x, y)	$G_{\mathrm{GE}}(x, y)$
3 adder	3A	$()$	$G_{3A}()$
addition	add	(m)	$G_{\mathrm{add}}(m)$
product	pro	(m, n)	$G_{\mathrm{pro}}(m, n)$
rank	rank	(m, n)	$G_{\mathrm{rank}}(m, n)$
Quantum random access	QRA	(m, n)	$G_{\mathrm{QRA}}()$
Superposition $(n \geq m)$	sp_m	$(B_{m,r})$	$G_{\mathrm{sp}_m}(B_{m,r})$
Superposition $(n < m)$	sp_n	$(B_{n,r})$	$G_{\mathrm{sp}_n}(B_{n,r})$
oracle phase flip	OPF	(V, M)	$G_{\mathrm{OPF}}(V, M)$
diffuser	dif	(V)	$G_{\mathrm{dif}}(V)$

Table 2. abbreviation, argument and denotion for each operation

fore, the quantum circuit that executes U_o has $\log|V|$ input qubits, no ancilla bit, the G-cost of $4\log|M|$ and the D-cost of 4.

5.8 The computational costs for diffuser

For the quantum state $|\psi\rangle \in \mathcal{H}^{\otimes|V|}$, let $x = \log|V|$, the diffuser can be constructed with one $C^{x-1}(Z)$ gate[3]. Therefore, the quantum circuit that performs U_d can be realized with $2(x-1)$ Toffoli gates and one Z gate. Hence, that quantum circuit has $\log|V|$ input qubits, $(\log|V|-1)$ ancilla bit, the G-cost of $48(\log|V|-1)-46 = 48\log|V|-94$ and the D-cost of 32.

5.9 The computational costs of quantum GRS algorithm

We discuss the computational costs of the quantum GRS algorithm and the improved version over the classical circuit and the quantum one. A detailed explanation is provided in APPENDIX B. **Table 2** shows the abbreviation, the argument and the denotion of the G-cost for each operation up to Section 5.8. That is, for each operation **op**, let **arg** be the argument of **op**, the most-right column in **Table 2** represents $G_{\mathbf{op}}(\mathbf{arg})$. Similarly, $D_{\mathbf{op}}(\mathbf{arg})$, $I_{\mathbf{op}}(\mathbf{arg})$ and $A_{\mathbf{op}}(\mathbf{arg})$ denote the D-cost, the number of the input qubits and the number of the ancilla bits for **op**. Also, **Table 3** shows the result up to the previous subsection. Here, $G_{\mathrm{GE}}(x, y) = (x-1)\{x(36y-20x+\frac{43}{2})+8\}$ and $D_{\mathrm{GE}}(x, y) = \frac{8}{3}(x-1)x(9y-4x+5)$. Also, '—' in **Table 3** denotes that there is no such qubits.

[3] For a positive integer x, $C^x(Z)$ gate means the quantum gate which makes Z gate on the target bit if x-controlled bits are $|1^x\rangle$. Also, $C^x(Z)$ gate can be realized with $2(x-1)$ Toffoli gates and one Z gate.

Operation	G-cost	D-cost	input	ancilla
GE	$G_{\mathrm{GE}}(x, y)$	$D_{\mathrm{GE}}(x, y)$	xy	$\frac{3}{2}(x-1)x$
3A	51	32	3	2
add	$26m - 23$	16	$2m$	$2m - 1$
pro	$\frac{9}{2}m(25m - 9)$	$16m$	$2m$	$m^2 + m + 1$
rank	$G_{\mathrm{GE}}(x, y) + xy$	$16(y + 1)$	xy	xy
QRA	$nm + n \log n$	$\log m + \log n$	$\log n + m + mn$	$nm + n \log n$
sp$_m$	$\|B_{m,r}\|$	1	$\|B_{m,r}\|$	—
sp$_n$	$\|B_{n,r}\|$	1	$\|B_{n,r}\|$	—
OPF	$4 \log \|M\|$	4	$\log \|V\|$	—
dif	$48 \log \|V\| - 94$	32	$\log \|V\|$	$\log \|V\| - 1$

Table 3. computational cost for each operation

5.9.1 The computational costs over the classical circuit

In estimating the computational costs of the classical circuit to simulate quantum GRS algorithm, we do not take account of the costs for the Grover's algorithm. Suppose $n \geq m$. We think about realizing the while sentence in **Algorithm 1** over the quantum circuit. In one iteration, $\mathrm{rank}_{\mathbb{F}_q}(e)$ is evaluated once, the multiplication of $H_{\ell,\ell'} F_j$ over $\mathbb{F}_{2^m}$ is $(n-k)nr$ times, the Gaussian Elimination for the $nr \times m(n-k)$ matrix over $\mathbb{F}_2$ is once, the multiplication of $\lambda_{\ell',j} F_j$ over $\mathbb{F}_{2^m}$ is $(n-k)nr$ times, and the addition over $\mathbb{F}_{2^m}$ is $(n-1)r$ times. These operations are repeated in ℓ_{QGRS} times.

5.9.2 The computational costs over the quantum circuit

To estimate the computational costs over the quantum circuit, we have to take account of four extra operations, the quantum random access, the superposition, the oracle process and the diffuser per one iteration in **Algorithm 5**, besides the computational costs in what we have just mentioned above.

5.10 Evaluation criteria and Parallelizing the quantum algorithm

NIST [20] states that the classical circuit corresponding to the 128, 192 and 256 security bits is equivalent to the classical circuit having the $2^{143}, 2^{207}$ and 2^{272} classical gates respectively. Also, the quantum circuit having the 128, 192 and 256 security level is equivalent to the quantum circuit whose the sum of the G-cost and the D-cost is 170, 233 and 298. Therefore, by considering a classical gate as a RAM operation by a classical computer, we can directly compare the G-cost with the above numbers. **Table 4** shows the security evaluation criteria simulated over classical and quantum circuits. For example, for a cryptosystem

security bit	classical circuit	quantum circuit
128	G-cost ≥ 143	G-cost + D-cost ≥ 170
192	G-cost ≥ 207	G-cost + D-cost ≥ 233
256	G-cost ≥ 272	G-cost + D-cost ≥ 298

Table 4. security criteria

with 128 security level, if its G-cost is greater than 143, it is secure against this attack.

Also, the D-cost is limited to 96 or less under the condition from NIST [20]. If the D-cost exceeds 96, we use the parallel Grover in [11]. This is the technique parallelizing **Algorithm 3**. Let G, D and W be the G-cost, D-cost and W-cost, respectively, for the entire statement with one processor. Then G-cost, D-cost and W-cost for the entire statement with p processors are $\sqrt{p}G$, $\frac{1}{\sqrt{p}}D$ and pW. Therefore, if D-cost exceeds 96 with one processor, D-cost can be reduced to less than 96 by using an appropriate number of processors for parallelizing.

5.11 Result

Table 5 lists the computational costs for each encryption scheme and security level. The upper row for each security level in the table shows the computational costs of the quantum GRS algorithm over the classical circuit. The middle row shows the computational costs of the quantum GRS algorithm over the quantum circuit. The lower row shows the computational costs of the improved quantum GRS algorithm over the quantum circuit. By the upper and middle rows, in the quantum GRS algorithm, we find the computational costs over the classical circuit are less than those over the quantum one. There is the difference between the middle row and the lower row because the size of $|V|$ is small in the improved quantum GRS algorithm. Also, **Table 5** shows that this attack method using the improved quantum GRS algorithm is effective for RQC. In addition, in RQC, the computational costs in security level 256 are less than those in security leve 192. One of the factors contributing to this is that the value of the parameter k in 256 security level is less than that in 192 security level. In general, for $r \in [w, n-k]$, $\binom{n}{r}_q = \binom{n}{n-r}_q$ holds. So, in **Table 1**, $\binom{n}{n-k}_q = \binom{n}{k}_q < \binom{n}{w}_q$ holds. Therefore, $\binom{n}{r}_q = \binom{n}{k}_q = \binom{179}{3}_2 \approx 2^{530}$ in 256 security level is smaller than $\binom{n}{r}_q = \binom{149}{5}_2 \approx 2^{722}$. Based on the above discussion, one reason why our attack method is effective for RQC is that the value of the parameter k is small. Then, **Table 6** shows the parameter k and the computational costs that RQC can be secure against this attack by the improved quantum GRS algorithm when the instances except for k are the same in **Table 1**.

Cryptosystem	Security bit	G-cost	D-cost	W-cost
ROLLO	128	478	95	658
		1344	95	1413
		448	95	258
	192	647	95	913
		1868	95	1980
		616	95	341
	256	894	95	1202
		2784	95	2958
		862	95	464
RQC	128	84	74	77
		492	60	492
		83	60	83
	192	107	95	101
		926	72	926
		116	72	116
	256	96	85	89
		764	67	764
		99	67	99

Table 5. Computational costs for each cryptosystem and security bit

Security bit	k	G-cost	D-cost	W-cost
128	5	104	67	104
192	8	152	84	152
256	11	215	95	223

Table 6. The proposal for RQC's parameter k

6 Conclusion

In this paper, for the rank CBC, we have proposed the quantum GRS algorithm, the best known algorithm for the rank SDP, and the improved version of that. Also, we have proposed the attack method with the quantum GRS algorithm for the rank CBCs remained at the 2nd Round of the NIST PQC standardization project. As a result, this attack method is effective for RQC, so we have recommended the value of k for RQC which is secure against this attack.

References

1. N. Aragon, P. Gaborit, A. Hauteville, J. P. Tillich: “A new algorithm for solving the rank syndrome decoding problem”, In The 2018 IEEE International Symposium on Information Theory (ISIT), pp.2421–2425, IEEE, 2018.
2. M. Bardet, P. Briaud et al.: “An algebraic attack on rank metric code-based cryptosystems”, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp.64–93, Springer, 2020.
3. A. Becker, A. Joux, A. May, A. Meurer: “Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding”, In Annual international conference on the theory and applications of cryptographic techniques, pp.520–536, Springer, 2012.
4. D. J Bernstein: “Grover vs. McEliece”, In International Workshop on Post-Quantum Cryptography, pp.73–80, Springer, 2010.
5. E. M. Gabidulin: “Theory of codes with maximum rank distance”, Problemy peredachi informatsii, vol.21, no.1, pp.3–16, 1985.
6. P. Gaborit, G. Murat et al.: “Low rank parity check codes and their application to cryptography”, In Proceedings of the Workshop on Coding and Cryptography WCC, vol.2013, 2013.
7. P. Gaborit, O. Ruatta, J. Schrek: “On the complexity of the rank syndrome decoding problem ” , IEEE Transactions on Information Theory, vol.62, no.2, pp.1006–1019, 2015.
8. P. Gaborit, G. Zémor: “On the hardness of the decoding and the minimum distance problems for rank codes”, IEEE Transactions on Information Theory, vol.62, no.12, pp.7245–7252, 2016.
9. R. Gallager: “Low-density parity-check codes”, IRE Transactions on information theory, vol.8, no.1, pp.21–28, 1962.
10. L. K. Grover: “A fast quantum mechanical algorithm for database search”, In Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing, pp.212–219, 1996.
11. S. Jaques, J. M. Schanck: “Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE”, In Annual International Cryptology Conference, pp.32-61, Springer, 2019.
12. D. S. Johnson: “A catalog of complexity classes”, In Algorithms and complexity, pp.67–161, Elsevier, 1990.
13. G. Kachigar, J. P. Tillich: “Quantum information set decoding algorithms”, In International Workshop on Post-Quantum Cryptography, Lecture Notes in Computer Science, vol.10346, pp.69–89, Springer, 2017.
14. F. Levy-dit-Vehel, L. Perret: “Algebraic decoding of rank metric codes”, Proceedings of YACC, pp.142–152, 2006.

15. A. May, A. Meurer, E. Thomae: "Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$", In International Conference on the Theory and Application of Cryptology and Information Security, pp.107–124, Springer, 2011.
16. R. J. McEliece: "A public-key cryptosystem based on algebraic coding theory", Coding Thv, vol.4244, pp.114–116, 1978.
17. C. A. Melchor, N. Aragon et al.: "Hauteville: Rank quasi cyclic (RQC)", In Second round submission to the NIST post-quantum cryptography call, 2019.
18. C. A. Melchor, N. Aragon et al.: "ROLLO – Rank–Ouroboros, LAKE & LOCKER", In Second round submission to the NIST post-quantum cryptography call, 2019.
19. M. A Nielsen, I. Chuang: "Quantum computation and quantum information", Cambridge University Press, 2002.
20. NIST: "Post-Quantum Cryptography, Security (Evaluation Criteria)", available at `https://csrc.nist.gov/projects`.
21. S. Perriello, A. Barenghi, G. Pelosi: "A complete quantum circuit to solve the information set decoding problem", In 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp.366–377, IEEE, 2021.
22. E. Prange: "The use of information sets in decoding cyclic codes", IRE Transactions on Information Theory, vol.8, no.5, pp.5–9, 1962.
23. I. S. Reed, G. Solomon: "Polynomial codes over certain finite fields", Journal of the society for industrial and applied mathematics, vol.8, no.2, pp.300–304, 1960.
24. V. V. Shende, I. L. Markov: "On the cnot-cost of toffoli gates", arXiv preprint arXiv:0803.2316, 2008.
25. P. W. Shor: "Algorithms for quantum computation: discrete logarithms and factoring", In Proceedings 35th annual symposium on foundations of computer science, pp.124–134, IEEE, 1994.

A The details for the computational costs of RQC

In moving only the parameter k, we show the tables (**Table 7-Table 9**) for the computational costs of 128, 192 and 256 security level RQC, respectively. In each table, if the value of k is in the lower row, the sum of the corresponding the G-cost and D-cost exceeds the criteria given in **Table 4**.

k	G-cost	D-cost	G-cost + D-cost
3	83	60	143
4	94	63	157
5	104	67	171

Table 7. The computational costs for 128 security level RQC

k	G-cost	D-cost	G-cost + D-cost
5	116	72	188
6	128	76	204
7	140	80	220
8	152	84	236

Table 8. The computational costs for 192 security level RQC

k	G-cost	D-cost	G-cost + D-cost
3	99	67	166
4	113	72	185
5	126	76	202
6	140	81	221
7	153	85	238
8	167	90	257
9	180	94	274
10	198	95	293
11	215	95	310

Table 9. The computational costs for 256 security level RQC

B Deriving the computational costs for the improved quantum GRS algorithm

We estimate the computational costs for the improved quantum GRS algorithm. We use the denotion of the computational costs given in **Table 2**.

B.1 In the case $n \geq m$

The operations from Line 4 to Line 18 in **Algorithm 8** are shown in **Table 10**. Here, in Line 17 and 18, $|V| = |V'_{m,r}|$ and $|M| = 1$. Also, $G_{\text{IGroloop},n\geq m}$, $D_{\text{IGroloop},n\geq m}$ and $A_{\text{IGroloop},n\geq m}$ denote the G-cost, D-cost and the number of the ancilla bits from Line 4 to Line 18 in **Algorithm 8**. Then,

$$\begin{aligned}
G_{\text{IGroloop},n\geq m} &= r(G_{\text{QRA}}(m, \log|V'_{m,r}|) + G_{\text{QRA}}(m,r)) + (n-k)nrG_{\text{pro}}(m) \\
&\quad + G_{\text{GE}}(m(n-k), nr) + nr(G_{\text{add}}(m) + G_{\text{pro}}(m)) \\
&\quad + G_{\text{rank}}(m,n) + G_{\text{OPF}}(V,M) + G_{\text{dif}}(V), \\
D_{\text{IGroloop},n\geq m} &= \max\{r \cdot \max\{D_{\text{QRA}}(m, \log|V'_{m,r}|), D_{\text{QRA}}(m,r)\} \\
&\qquad , (n-k)nrD_{\text{pro}}(m), D_{\text{GE}}(m(n-k), nr), nr \cdot D_{\text{add}}(m) \\
&\qquad , nr \cdot D_{\text{pro}}(m), D_{\text{rank}}(m,n), D_{\text{OPF}}(V,M), D_{\text{dif}}(V)\}
\end{aligned}$$

Line	operation
6	$\mathrm{QRA}(m, \log\lvert V'_{m,r}\rvert)$ and $\mathrm{QRA}(m, r)$
10	$\mathrm{pro}(m)$
13	$\mathrm{GE}(m(n-k), nr)$
15	$\mathrm{pro}(m)$ and $\mathrm{add}(m)$
16	$\mathrm{rank}(m, n)$
17	$\mathrm{OPF}(V, M)$
18	$\mathrm{dif}(V)$

Table 10. each operation in Improved Grover$_{n\geq m}$

$$\begin{aligned}A_{\mathrm{IGroloop},n\geq m} &= r(A_{\mathrm{QRA}}(m, \log\lvert V'_{m,r}\rvert) + A_{\mathrm{QRA}}(m, r)) + (n-k)nrA_{\mathrm{pro}}(m)\\ &\quad + A_{\mathrm{GE}}(m(n-k), nr) + nr(A_{\mathrm{add}}(m) + A_{\mathrm{pro}}(m))\\ &\quad + A_{\mathrm{rank}}(m,n) + A_{\mathrm{OPF}}(V, M) + A_{\mathrm{dif}}(V).\end{aligned}$$

The for sentence is executed in $\ell_{\mathrm{QGRS},n\geq m}$ times. $G_{\mathrm{IGro},n\geq m}, D_{\mathrm{IGro},n\geq m}$ and $A_{\mathrm{IGro},n\geq m}$ denote the G-cost, D-cost and the number of the ancilla bits in **Algorithm 8**. Hence, p_{Igro} processors are in parallel, the following holds:

$$\begin{aligned}G_{\mathrm{IGro},n\geq m} &= G_{\mathrm{IGroloop},n\geq m} \cdot \ell_{\mathrm{QGRS},n\geq m} \cdot \sqrt{p_{\mathrm{Igro}}} + G_{\mathrm{QRA}}(m, \log\lvert V'_{m,r}\rvert),\\ D_{\mathrm{IGro},n\geq m} &= \max\{D_{\mathrm{IGroloop},n\geq m} \cdot \ell_{\mathrm{QGRS},n\geq m} \cdot \frac{1}{\sqrt{p_{\mathrm{Igro}}}}, D_{\mathrm{QRA}}(m, \log\lvert V'_{m,r}\rvert)\},\\ A_{\mathrm{IGro},n\geq m} &= A_{\mathrm{IGroloop},n\geq m} \cdot \ell_{\mathrm{QGRS},n\geq m} \cdot p_{\mathrm{Igro}} + A_{\mathrm{QRA}}(m, \log\lvert V'_{m,r}\rvert).\end{aligned}$$

Therefore, $G_{\mathrm{IQGRS},n\geq m}, D_{\mathrm{IQGRS},n\geq m}$ and $A_{\mathrm{IQGRS},n\geq m}$ denote the G-cost, D-cost and the number of the ancilla bits in **Algorithm 9**. Then,

$$\begin{aligned}G_{\mathrm{IQGRS},n\geq m} &= G_{\mathrm{IGro},n\geq m} + (n-k)nrG_{\mathrm{pro}}(m)\\ &\quad + G_{\mathrm{GE}}(m(n-k), nr) + nr(G_{\mathrm{add}}(m) + G_{\mathrm{pro}}(m)),\\ D_{\mathrm{IQGRS},n\geq m} &= \max\{D_{\mathrm{IGro},n\geq m}, (n-k)nrD_{\mathrm{pro}}(m)\\ &\qquad , D_{\mathrm{GE}}(m(n-k), nr), nr\max\{D_{\mathrm{add}}(m), D_{\mathrm{pro}}(m)\},\\ A_{\mathrm{IQGRS},n\geq m} &= A_{\mathrm{IGro},n\geq m} + (n-k)nrA_{\mathrm{pro}}(m)\\ &\quad + A_{\mathrm{GE}}(m(n-k), nr) + nr(A_{\mathrm{add}}(m) + A_{\mathrm{pro}}(m)).\end{aligned}$$

In addition, $I_{\mathrm{IGro},n\geq m}$ and $W_{\mathrm{IGro},n\geq m}$ denote the number of input qubits and the W-cost in **Algorithm 9**, respectively. Here, $I_{\mathrm{IGro},n\geq m} = m(n-k)\cdot mn + m(n-k) + m^2 + \lvert B'_{m,r}\rvert \cdot m^2$ because **Algorithm 9** is the algorithm for H, s, β and $B'_{m,r}$. We estimate the W-cost by $W_{\mathrm{IGro},n\geq m} = I_{\mathrm{IGro},n\geq m} + A_{\mathrm{IGro},n\geq m}$.

B.2 In the case $n < m$

We can compute the same in this case. Improved_Grover$_{n<m}$ denote the subroutine using the Grover's algorithm in this case, and that algorithm is given in **Algorithm 13**.

Algorithm 13 Improved_Grover$_{n<m}$

```
Input: $w, H, s, \beta, B'_{n,r}$
Output: $F'$
1: $|\psi\rangle \leftarrow |0^{|B'_{n,r}|}\rangle$
2: $|\psi\rangle \leftarrow H^{\otimes|B'_{n,r}|}|\psi\rangle$
3: for $x := 1$ to $\sqrt{\frac{\binom{n}{w}_q}{\binom{r}{w}_q}}$ do
4:     $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
5:     for $j := 1$ to $r$ do
6:         $|F'_{\psi,j}\rangle \leftarrow \mathrm{QRA}((\mathrm{QRA}(|B_{n,r}\rangle, |\psi\rangle)), |j\rangle)$
7:     for $\ell := 1$ to $n-k$ do
8:         for $i := 1$ to $m$ do
9:             for $i' := 1$ to $m$ do
10:                for $j := 1$ to $r$ do
11:                    $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |0\rangle^{\lceil \log_2 q \rceil}$
12:                    for $\ell' := 1$ to $n$ do
13:                        if $\hat{\imath} <= n$ then
14:                            $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,\hat{\imath}}\rangle) \cdot |\beta_{i'}\rangle$
15:                        else
16:                            $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,n}\rangle) \cdot |\beta_{i'}\rangle$
17:    for $\ell := 1$ to $n-k$ do
18:        $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
19:    $(|\lambda_{\ell',j}\rangle)_{1\le\ell'\le n, 1\le j\le r} \leftarrow \mathsf{GE}(|\hat{H}'\rangle, |\hat{s}\rangle)$
20:    for $i := 1$ to $m$ do
21:        $|e'_i\rangle \leftarrow |0^{\log q}\rangle$
22:        for $j := 1$ to $r$ do
23:            $|e'_i\rangle \leftarrow |e'_i\rangle + |\lambda'_{\ell',j}\rangle \cdot |F'_{\psi,j}\rangle$
24:        for $\ell' := 1$ to $n$ do
25:            $|e_{\ell'}\rangle \leftarrow |e_{\ell'}\rangle + |e'_{i,\ell'}\rangle|\beta_{\ell'}\rangle$
26:    if $\mathrm{rank}_{\mathbb{F}_q}(|e\rangle)! = w$ then
27:        $|\psi\rangle \leftarrow -|\psi\rangle$
28:    $|\psi\rangle \leftarrow U_d|\psi\rangle$
29: return $\mathrm{QRA}(|B_{n,r}\rangle, |\psi\rangle)$
```

Algorithm 14 Improved quantum GRS algorithm （$n < m$）

```
Input: $q, m, n, k, w, H, s, \beta = (\beta_1, \cdots, \beta_m), r, B'_{n,r}$
Output: $e$
 1: $|e\rangle = |e_1\rangle \cdots |e_r\rangle \leftarrow |0^{\log q}\rangle \cdots |0^{\log q}\rangle$
 2: $|F'\rangle = |F'_1\rangle \cdots |F'_r\rangle \leftarrow \mathsf{Grover}_{n<m}(w, H, s, \beta, B'_{n,r})$
 3: for $\ell := 1$ to $n - k$ do
 4:   for $i := 1$ to $m$ do
 5:     for $i' := 1$ to $m$ do
 6:       for $j := 1$ to $r$ do
 7:         $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |0\rangle^{\lceil \log_2 q \rceil}$
 8:         for $\ell' := 1$ to $n$ do
 9:           if $\hat{i} <= n$ then
10:             $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,\hat{i}}\rangle) \cdot |\beta_{i'}\rangle$
11:           else
12:             $|(\hat{H}'_{\ell,i})_{j,i'}\rangle \leftarrow |(\hat{H}'_{\ell,i})_{j,i'}\rangle + p_{i'}(|H_{\ell,n}\rangle) \cdot |\beta_{i'}\rangle$
13: for $\ell := 1$ to $n - k$ do
14:   $|\hat{s}_\ell\rangle \leftarrow |p_1(|s_\ell\rangle)\rangle \cdots |p_m(|s_\ell\rangle)\rangle$
15: $(|\lambda_{\ell',j}\rangle)_{1 \le \ell' \le n, 1 \le j \le r} \leftarrow \mathsf{GE}(|\hat{H}'\rangle, |\hat{s}\rangle)$
16: for $i := 1$ to $m$ do
17:   $|e'_i\rangle \leftarrow |0^{\log q}\rangle$
18:   for $j := 1$ to $r$ do
19:     $|e'_i\rangle \leftarrow |e'_i\rangle + |\lambda'_{\ell',j}\rangle \cdot |F'_j\rangle$
20:   for $\ell' := 1$ to $n$ do
21:     $|e_{\ell'}\rangle \leftarrow |e_{\ell'}\rangle + |e'_{i,\ell'}\rangle|\beta_{\ell'}\rangle$
22: return $|e\rangle$
```

$G_{\text{IGroloop},n<m}, D_{\text{IGroloop},n<m}$ and $A_{\text{IGroloop},n<m}$ denote the G-cost, D-cost and the number of the ancilla bits from Line 4 to Line 28 in **Algorithm 13**. Also, $G_{\text{IGro},n<m}, D_{\text{IGro},n<m}$ and $A_{\text{IGro},n<m}$ denote the G-cost, D-cost and the number of the ancilla bits in **Algorithm 13**, and we can estimate these computational costs by $G_{\text{IGroloop},n<m}, D_{\text{IGroloop},n<m}, A_{\text{IGroloop},n<m}, \ell_{\text{QGRS},n\ge m}$ and p'_{Gro}, where p'_{Gro} means the number of processors in parallel.

$G_{\text{IQGRS},n<m}, D_{\text{IQGRS},n<m}$ and $A_{\text{IQGRS},n<m}$ denote the G-cost, D-cost and the number of the ancilla bits in the improved quantum GRS algorithm when $n < m$（**Algorithm 14**）. We can evaluate these costs using $G_{\text{IGro},n<m}, D_{\text{IGro},n<m}$ and $A_{\text{IGro},n<m}$, and compute the W-cost in the same way.