Quantum Gauss-Jordan Elimination for Code in Quantum

1st Kyungbae Jang
IT Department
Hansung University
Seoul, South Korea
starj1023@gmail.com

2nd Hyunji Kim
IT Department
Hansung University
Seoul, South Korea
khj1594012@gmail.com

3th Hwajeong Seo
IT Department
Hansung University
Seoul, South Korea
hwajeong84@gmail.com

***Abstract*—Quantum computers can efficiently model and solve certain problems on their own. Already in various fields, quantum computing is expected to outperform classical computers. In this flow, it is important to quantize classical arithmetic to achieve the benefits of quantum computing. In this paper, we efficiently implement quantum Gauss-Jordan elimination for binary matrix. Quantum Gauss-Jordan is required to accelerate Information Set Decoding, a cryptanalysis algorithm for code-based post-quantum ciphers, with Grover's algorithm. In our understanding, the most important module when implementing ISD as a quantum version is quantum Gauss-Jordan elimination. We implement quantum Gauss-Jordan using only quantum gates (e.g., X, CX, CCX, and Swap) that replace classical operations. Finally, we analyze the quantum resources required for our implementation. For the simulation and resource estimation of our work, the quantum programming tool ProjectQ is used.**

***Index Terms*—Quantum computer, Gauss-Jordan elimination, ISD, Code-based, Grover's algorithm**

I. Introduction

In some fields of computer science, quantum computers using quantum mechanics are expected to provide a different level of computing power than classical computers. Such quantum computing power effectively solves problems that classical computers cannot solve. Current cryptosystems are designed with the challenges of classical computers, but quantum computers are considered to effectively solve these challenges. As the era of high-level quantum computing approaches, combining quantum with cryptanalysis [1]–[4] is an active area of research.

To take advantage of the advantages of quantum computing, it is important to efficiently port classical arithmetic to the quantum domain. With these motives, studies are being proposed to optimize cryptographic arithmetic operations as quantum circuits [5]–[9]. In this paper, we present quantum Gauss-Jordan elimination targeting binary matrices, a core module of quantum cryptanalysis for code-based cryptography. Previously, quantum Gauss-Jordan elimination was presented in [10]. They implement Gauss-Jordan elimination using Grover search algorithm [11]. However, we present quantum Gauss-Jordan elimination without using Grover's algorithm. Our implementation only uses multi-controlled swap gates, X, CX, and multi-controlled X gates. Our work is effective for quantization of Information Set Decoding (ISD), one of the cryptanalysis algorithms for code-based cryptography.

A. Our Contribution

Our contributions are as follows:

1) **Quantum Gauss-Jordan elimination for binary matrix.** We present a quantum version of Gauss-Jordan elimination for finding the inverse of a binary matrix. Quantum Gauss-Jordan elimination can be used in various ways in quantum computing. Among them, we focus on implementing a quantum version of ISD, a cryptanalysis technique for code-based cryptography.
2) **Efficient quantum implementation of Gauss-Jordan elimination**. In the previous implementation of quantum Gauss-Jordan, the Grover algorithm is used. Grover's algorithm requires a lot of resources because iteration of the quantum circuit is essential. On the other hand, we efficiently implement quantum Gauss-Jordan using only quantum gates that replace classical operations (XOR, AND, Swap).

II. Background

A. Gauss-Jordan Elimination for Code

In code-based cryptography, the security is based on the difficulty of the syndrome decoding problem. Let the parity check matrix H be the public key (i.e., Niederreiter system). The vector m of weight-t (secret) is multiplied by H and a syndrome c (ciphertext) is generated (i.e., $H \cdot m^T = c$).

Information Set Decoding (ISD) is a well-known algorithm for solving the syndrome decoding problem. In summary, ISD can be treated as a brute force attack that reduces the search space. The steps for ISD are as follows. An information set S is randomly extracted from a matrix H. If the matrix S is invertible, check the weight condition by multiplying S^{-1} by the syndrome c. If the weight of the result vector is t, the attacker can recover m. Otherwise, go back to the beginning and repeat.

Gauss-Jordan elimination can compute the inverse of an invertible matrix. So it can be used to compute S^{-1}, which is the inverse of the information set S. Gauss-Jordan elimination computes an inverse matrix through operations between rows to make the target matrix an identity matrix.

Actually, in ISD, the final vector to compute is $S^{-1} \cdot c^T$, which can be computed without generating S^{-1}. If we perform

the operations used to perform Gauss-Jordan elimination on S directly on c, we can compute a result vector equal to $S^{-1} \cdot c^T$. Simply put, we do not construct S^{-1}, but apply only the operations to construct S^{-1} to syndrome c.

B. Quantum Gates

In this section, quantum gates for implementing Gauss-Jordan elimination as a quantum circuit are described. There are representative quantum gates that can replace classical operations.

The quantum gate CX (Figure 1) can replace the classic XOR operation by inverting the value of the target qubit when the control qubit is 1. Another quantum gate, CCX (Figure 2), can replace the classic AND operation by inverting the value of the target qubit if both control qubits are 1. Actually, the CCX gate is implemented as a combination of CX gates and single gates, so the cost is relatively high. There are several ways to implement CCX gates. A quantum Swap gate (Figure 3) exchanges the state of two qubits.

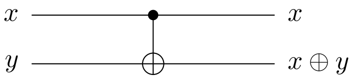

Fig. 1. CX gate.

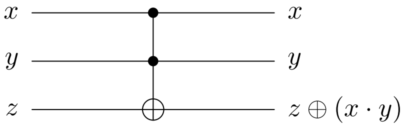

Fig. 2. CCX gate.

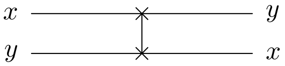

Fig. 3. Swap gate.

C. Quantum Gauss-Jordan Elimination

III. Proposed Quantum Gauss-Jordan Elimination

Quantum Gauss-Jordan elimination was first introduced by ABC et al. in [10]. In their work, the Grover algorithm is used to perform quantum Gauss-Jordan elimination. However, we present quantum Gauss-Jordan using only quantum gates without using Grover's algorithm. Also, we target Gauss-Jordan elimination for binary matrices. In Quantum Information Set Decoding (QISD) [12]–[14], a Grover oracle is designed to find a solution in an information set in a superposition state.

Re-applying Grover search for quantum Gauss-Jordan elimination in Grover oracle does not reduce the complexity of classical ISD to the square root. On the other hand, our quantum Gauss-Jordan elimination is suitable for QISD because it is implemented only with quantum arithmetic.

Actually, in ISD, the final vector to compute is $S^{-1} \cdot c^T$, which can be computed without generating S^{-1}. If we perform the operations used to perform Gauss-Jordan elimination on S directly on c, we can compute a result vector equal to $S^{-1} \cdot c^T$. Simply put, we do not construct S^{-1}, but apply only the operations to construct S^{-1} to syndrome c. So, we apply this approach to optimize our quantum implementation.

A. Prepare Input using Butterfly Network

To prepare the information set in the superposition state, we adopt the Butterfly Network (BN) approach introduced in [15].

We can use this BN to satisfy an n-qubit vector with superposition state that satisfies weight = k. We use the vector we created using BN to generate the permutation matrix. Actually, new qubits have to be allocated to create the permutation matrix. Then we need to check the controlled-swap gates column by column and generate the correct permutation matrix according to the position of 1. However, since this paper focuses on the quantum Gauss-Jordan, we do not go into the details of generating the permutation matrix.

We can obtain the information set in the superposition state by multiplying the generated permutation matrix by the set public key H. This can be implemented intuitively using only CCX gates. Now we can apply our quantum Gauss-Jordan elimination to compute the vector multiplied by the inverse of the information set and the syndrome (i.e., $S^{-1} \cdot c^T$).

The proposed method converts the information set into an identity matrix by performing elimination and arrange column by column using the temp column. In our implementation we need $2 \cdot k$ ancilla qubits for temp columns. Our primary goal is to transform the bottom diagonal matrix into an identity form as shown in Figure 4, which means that the information set is invertible.

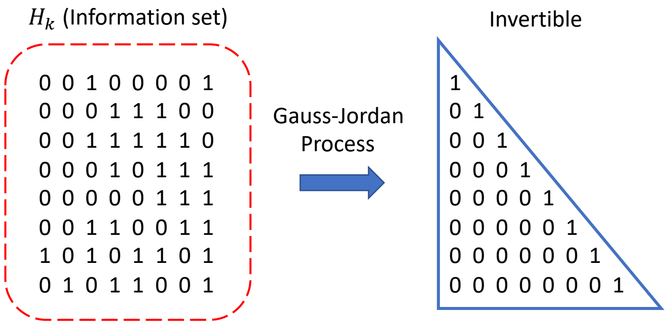

Fig. 4. Primary goal of Gauss-Jordan elimination.

The process is as follows. Copy the first column of the information set to the temp column (k ancilla qubits) using k CX gates. Then, elimination is performed on the target column of the information set. That is, if there are multiple 1's in the column, only one is left and the others are eliminated. The quantum circuit design for elimination is described in Algorithm 1.

```
Algorithm 1 Quantum implementation of elimination.
Input: $S$, $c$, $l$-th Column $col_l$ of $S$ ($k$ qubits), and temp column $t$ ($k$ ancilla qubits)
Output: Eliminated $l$-th column $col_l$, $S$, and $c$
for $i = 0$ to $(k - l - 2)$ do
    for $j = 0$ to $(k - l - 2 - i)$ do
        CCX($col_l[l+i]$, $t[l+i+j+1]$, $col_l[l+i+j+1]$)
        CCCX($col_l[l+i]$, $t[l+i+j+1]$, $c[l+i]$), $c[l+i+j+1]$
        for $p = 0$ to $(k - l - 2)$ do
            CCCX($col_l[l+i]$, $t[l+i+j+1]$, $col_{l+p+1}[l+i]$), $col_{l+p+1}[l+i+j+1]$
        end for
    end for
end for
return $col_l$, $S$, and $c$
```

We perform elimination according to the target column, col_l, and apply these operations including other columns of the information set (i.e., from col_{l+1} to col_{k-1}). As mentioned earlier, our implementation computes $S^{-1} \cdot c^T$ rather than generating S^{-1}. Thus, elimination performed according to the target column is equally applied to syndrome c. There are still steps left, but c will change to $S^{-1} \cdot c^T$ when quantum Gauss-Jordan is finished.

After the elimination step, we rotate the target column to form an identity matrix. This ensures that the value 1 of the target column is in the correct place (the values on the diagonal of the information set must be 1). We can be reminded by going to Figure 4. An overview of the arrange step is shown in Figure 5.

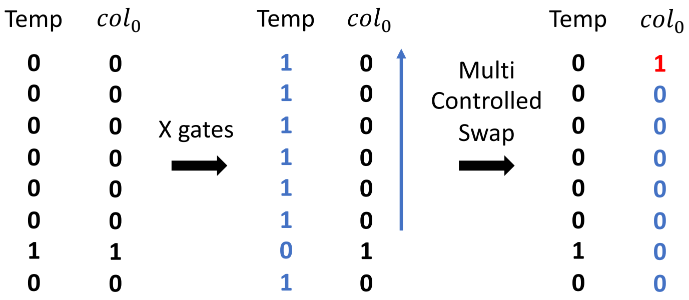

Fig. 5. Overview of the arrange step.

At the beginning of the arrange step, X gates are performed on the entire temp column. Then, the values of the temp column are inverted. Now, we use the temp column to position the value 1 of col_l to the appropriate position. For this purpose, sequentially increasing multi-controlled swap gates are used. The qubits of the temp column become the control qubits, and col_l is rotated. Although only col_l is shown in Figure 5, the arrange operation is applied equally to information set S and syndrome c. The details of the arrange step are described in Algorithm 2. Multi(A)-Controlled Rotation(B, C) of Algorithm 2 means that the qubits of B act as control qubits, so that if B is all 1, the rotation of Algorithm 3 is performed on column C. Algorithm 3 arranges one column based on the temp column. Thus, it should be applied to all the right columns of col_l and also to syndrome c.

```
Algorithm 2 Quantum implementation of Arrange.
Input: $k$, $l$, target column $col_t$, and temp column $t$
Output: $col_t$ (arranged)
for $i = 0$ to $(k - l - 2)$ do
    Multi($i+1$)-Controlled Rotation($t[l \sim (l+i)]$, $col_t$)
end for
return $col_t$
```

```
Algorithm 3 Quantum implementation of Rotation.
Input: $k$, $l$, and $col_t$
Output: $col_t$ (rotated)
for $i = 0$ to $(k - l - 2)$ do
    Swap($col_t[(l+i)$, $col_t[(l+i+1)]$
end for
return $col_t$
```

Suppose that elimination and arrange for the first column are completed in the proposed quantum Gauss-Jordan elimination. The first column can be used as new ancilla qubits because all values except the first row are currently 0. Since the first column is no longer needed in process (We have already applied all the operations corresponding to the first column), we replace the first column with the temp column. In Gauss-Jordan elimination, since the second column can ignore the value for the first row, the value from the second row is copied to the new temp column. In this way, qubits for the temp column are saved and arrange and elimination are performed up to the last column.

As a result, all the operations for transforming the lower triangular matrix into an identity matrix are applied to syndrome c.

Now we perform the elimination step in the opposite direction, i.e., removing 1's in the upper triangular matrix (no need to perform arrange since the diagonal values are all 1). We allocate k ancilla qubits for the temp column to perform elimination. However, in the same way as before, the column on which elimination has been performed replaces the temp column. Thus, new qubits for temp column are not allocated thereafter.

If the elimination step in the opposite direction is completed, syndrome c changes to a value of $S^{-1} \cdot c^T$. The proposed method applies Gauss-Jordan's operations to syndrome c in an on-the-fly approach without generating an inverse matrix of the information set. This on-the-fly approach effectively reduces qubits, gates, and depth.

The last step in QISD checks if the weight of the vector $S^{-1} \cdot c^T$ is t. If weight is t, $S^{-1} \cdot c^T$ tells us the error positions of the secret (i.e., positions of 1). We can use the weight check module using quantum adders introduced in [15]. It starts with the addition of 1-qubit units of $S^{-1} \cdot c^T$, and the weight is computed in the final addition.

IV. Performance

In this section, we estimate the quantum resources required for the proposed quantum Gauss-Jordan elimination. We conduct simulations and resource estimation in ProjectQ [16], a quantum programming tool. Out implementation is verified by checking the result vector of quantum Gauss-Jordan using `ClassicalSimulator`, one of the libraries of ProjectQ. Another library, `ResourceCounter`, is used, which estimates the qubits, gates and depth required for quantum Gauss-Jordan. We estimate the quantum resources required for quantum Gaussian elimination for the information set H_k of matrix size (8×8, i.e., $k = 8$).

Our implementation needs k^2 qubits for the information set, k qubits for syndrome c, and $2 \cdot k$ qubits for the temp column. For quantum gates, we need X, multi-controlled CX gates and multi-controlled swap gates. In our implementation, the process of checking one row and one column for a wide range of the target matrix is repeated to perform quantum Gauss-Jordan. We believe that this iterative inspection process is an unconditional choice. Although the depth is not low, it will be more efficient than performing Grover iteration as in [10] (There are no reports of required quantum resources).

TABLE I
Quantum resources required for our Gauss-Jordan elimination.

H_k size	Qubits	#X	#CX	#CCX	#CCCX	#Multi-Controlled Swap	Full Depth
8×8	88	56	70	140	546	1,064	1,404

V. Conclusion

In this paper, we present an efficient quantum circuit implementation for Gauss-Jordan elimination. The difference from previous research is that we target binary matrices and are implemented without Grover's algorithm. Actually, as observed in the quantum resources used in our implementation, quantum Gauss-Jordan elimination is not a simple operation on a quantum computer. However, the circuit complexity increase due to Grover iteration can be avoided by implementing it using only quantum gates that replace classical operations. The quantum Gauss-Jordan elimination we implemented can be utilized in many fields in quantum computing (QISD as a representative example).

This time, we only focused on quantum Gauss-Jordan elimination, one of the modules, but the future work is to optimize the whole process of QISD. This will also contribute to quantum cryptanalysis for code-based post-quantum ciphers.

VI. Acknowledgement

This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00264, Research on Blockchain Security Technology for IoT Services, 25%) and this work was partly supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (<Q|Crypton>, No.2019-0-00033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity, 50%) and this work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00627, Development of Lightweight BIoT technology for Highly Constrained Devices, 25%).

References

[1] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing grover oracles for quantum key search on AES and LowMC," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 280–310, Springer, 2020.

[2] J. Zou, Z. Wei, S. Sun, X. Liu, and W. Wu, "Quantum circuit implementations of AES with fewer qubits," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 697–726, Springer, 2020.

[3] K. Jang, A. Baksi, G. Song, H. Kim, H. Seo, and A. Chattopadhyay, "Quantum analysis of aes," *Cryptology ePrint Archive*, 2022.

[4] T. Häner, M. Roetteler, and K. M. Svore, "Factoring using $2n + 2$ qubits with Toffoli based modular multiplication," *arXiv preprint arXiv:1611.07995*, 2016.

[5] D. Cheung, D. Maslov, J. Mathew, and D. K. Pradhan, "On the design and optimization of a quantum polynomial-time attack on elliptic curve cryptography," in *Workshop on Quantum Computation, Communication, and Cryptography*, pp. 96–104, Springer, 2008.

[6] S. Kepley and R. Steinwandt, "Quantum circuits for $\mathbb{F}_{2^n}$-multiplication with subquadratic gate count," *Quantum Information Processing*, vol. 14, no. 7, pp. 2373–2386, 2015.

[7] I. Van Hoof, "Space-efficient quantum multiplication of polynomials for binary finite fields with sub-quadratic Toffoli gate count," *arXiv preprint arXiv:1910.02849*, 2019.

[8] K. Jang, S. J. Choi, H. Kwon, Z. Hu, and H. Seo, "Impact of optimized operations $A \cdot B$, $A \cdot C$ for binary field inversion on quantum computers," in *International Conference on Information Security Applications*, pp. 154–166, Springer, 2020.

[9] K. Jang, G. J. Song, H. Kim, H. Kwon, W.-K. Lee, Z. Hu, and H. Seo, "Binary field montgomery multiplication on quantum computers," *Cryptology ePrint Archive*, 2021.

[10] D. N. Diep, D. H. Giang, and N. Van Minh, "Quantum gauss-jordan elimination and simulation of accounting principles on quantum computers," *International Journal of Theoretical Physics*, vol. 56, no. 6, pp. 1948–1960, 2017.

[11] 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.

[12] G. Kachigar and J.-P. Tillich, "Quantum information set decoding algorithms," in *International Workshop on Post-Quantum Cryptography*, pp. 69–89, Springer, 2017.

[13] E. Kirshanova, "Improved quantum information set decoding," in *International Conference on Post-Quantum Cryptography*, pp. 507–527, Springer, 2018.

[14] D. J. Bernstein, "Grover vs. mceliece," in *International Workshop on Post-Quantum Cryptography*, pp. 73–80, Springer, 2010.

[15] S. PERRIELLO, "Design and development of a quantum circuit to solve the information set decoding problem," 2019.

[16] D. S. Steiger, T. Häner, and M. Troyer, "Projectq: an open source software framework for quantum computing," *Quantum*, vol. 2, p. 49, 2018.