Quantum Gauss-Jordan Elimination for Code in Quantum

Kyungbae Jang, Hyunji Kim, and **Hwajeong Seo**

HANSUNG UNIVERSITY 한성대학교

CryptoCraft LAB

Contents

Our Contribution

- **We propose quantum Gauss-Jordan elimination**
  - **Implemented only with quantum gates**
  - **without Grover's algorithm**
- **Efficient quantum Gauss-Jordan elimination for binary matrix**
- **Quantum arithmetic for quantum cryptanalysis of code-based ciphers**

Code-based Cryptography

- **Applying coding theory to public key cryptosystems**
  - Generates a pair of **generation matrix** and **parity check matrix** from the defined code
    - Used for Encryption and Decryption

$Code$

Genrator Matrix $\boldsymbol{G}$ $\longrightarrow$ $\boldsymbol{mG} + \boldsymbol{e} = \boldsymbol{c}$

Parity Check Matrix $\boldsymbol{H}$ $\longrightarrow$ $\boldsymbol{Hc} = \boldsymbol{mGH} + \boldsymbol{eH} = \boldsymbol{eH}$

NIST Post-Quantum Cryptography Standardization

- **Post-quantum cryptography standardization contest in progress with NIST**
  - **In addition, round 4 is currently in progress**

NIST

Search CSRC

CSRC MENU

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

CSRC

UPDATES

2022

PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates

PQC Fourth Round Candidate Key-Establishment Mechanisms (KEMs)

The following candidate KEM algorithms will advance to the fourth round:

Public-Key Encryption/KEMs	
BIKE	Code-based!!
Classic McEliece	
HQC	
SIKE	Isogeny-based

Classic McEliece

- **Classic McEliece is a Niederreiter system that uses a parity check matrix as its public key**
  - **A randomly generated vector with a weight condition $\boldsymbol{t}$ is the secret value**

$$e = (0\ 0\ 0\ 0\ 0\ 0\ \mathbf{1}\ \mathbf{1}\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0) \rightarrow \textit{Secret}\ \textbf{(Weight } t = 2\textbf{)}$$

$$H = \left(\begin{array}{cccccccc|cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}\right) \rightarrow \textbf{Public key (Parity check matrix)}$$

- ***Challenge***

$$C = He = (00000011),\ e = ?$$

Information Set Decoding (ISD)

- ***Challenge***

$C = He = (00000011),\ e = ?$

$e = (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ \mathbf{1}\ \ \mathbf{1}\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)$

$$H = \left(\begin{array}{cccccccc|cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}\right)$$

※ ISD Summary

1. **Randomly select as many columns as the number of rows in the public key**
2. **The matrix constructed in this way is an information set, if the information set is invertible? Perform Gaussian Elimination**
3. **We can compute the inverse matrix of the information set → Information set^{-1}**
4. **Check $C \times$ Information set^{-1} 's weight**
5. **If we included all error locations in step 1, the attack succeeded, in case ↑, the attack failed in case ↑, the attack succeeded.**

Information Set Decoding (ISD)

- **When the result vector of [Generated Inverse X Ciphertext] is Weight t=2,**
  - **Attack success** → The result vector tells us the **positions of the values 1.**

$$\underset{(H_{n-k})^{-1}}{\begin{pmatrix} 1&1&0&1&1&0&1&0 \\ 1&1&0&1&0&1&0&1 \\ 1&1&1&1&1&1&0&0 \\ 0&0&0&1&1&0&0&0 \\ 1&0&1&1&0&0&0&0 \\ 1&1&1&0&1&0&0&0 \\ 1&0&0&1&1&1&0&0 \\ 0&1&1&1&1&1&0&0 \end{pmatrix}} \times \underset{C^T}{\begin{bmatrix} 0\\0\\0\\0\\0\\0\\1\\1 \end{bmatrix}} = \begin{bmatrix} 1\\1\\0\\0\\0\\0\\0\\0 \end{bmatrix} \rightarrow \text{Weight} = 2(t)$$

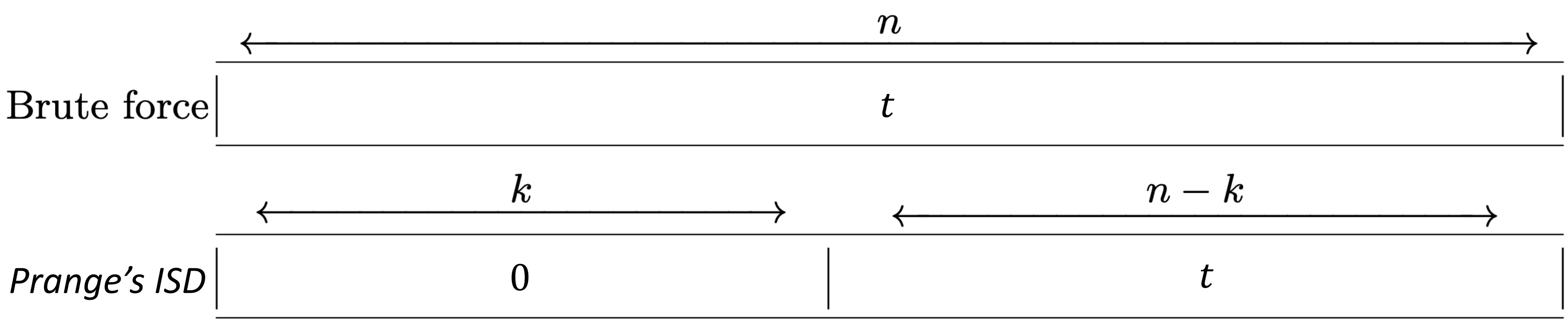
- Means that the **1st and 2nd of the column** selected in step 1 are 1 → **Recover *Secret***

e = 0 0 0 0 0 0 $\boxed{1\ 1\ 0\ 0\ 0\ 0\ 0\ 0}$ 0 0

Information Set Decoding (ISD) in Quantum

- **Information Set Decoding is an attack algorithm that reduces the search space of brute force**
- **Efficient Brute force(ISD) → Acceleration using the Grover algorithm**

  → Reduced complexity of square root ($\sqrt{\ }$)



Quantum Information Set Decoding (QISD)

- **Overbeck–Sendrier's analysis for QISD [1]**
  - **Grover's algorithm cannot reduce the complexity of information set decoding to the square root**

McEliece parameters m, t	Workload Cryptanalysis (in binary operations) classic	quantum computer
$11, 32$	2^{91}	2^{86}
$11, 40$	2^{98}	2^{94}
$12, 22$	2^{93}	2^{87}
$12, 45$	2^{140}	2^{133}

[1] R. Overbeck, N. Sendrier, "Code-based cryptography", 2009

Quantum Information Set Decoding (QISD)

- **Bernstein's QISD analysis in "Grover vs McEliece" paper[2]**
  - **Grover's algorithm can reduce the complexity of Information Set Decoding to the square root**
    - **But, on a theoretical level**

Author(s)	Year	$\max_{0\le R\le 1} \alpha(R,\omega_{\mathrm{GV}})$ to 4 dec. places
Prange [23]	1962	0.1207
Dumer [11]	1991	0.1164
MMT [18]	2011	0.1114
BJMM [4]	2012	0.1019
MO [19]	2015	0.0966

[3]

Reduced to the square root

Author(s)	Year	Ingredients	$\max_{0\le R\le 1} \alpha(R,\omega_{\mathrm{GV}})$ to 5 dec. places
Bernstein [5]	2010	Prange+Grover	0.06035
This paper	2017	Shamir-Schroeppel+Grover+QuantumWalk	0.05970
This paper	2017	MMT+"1+1=0"+Grover+QuantumWalk	0.05869

[3]

[2] D. J. Bernstein, "Grover vs. McEliece", PQCrypto, 2010
[3] G. Kachigar et al. "Quantum Information Set Decoding", Cryptography and Security (cs.CR); Quantum Physics, 2017

Quantum Information Set Decoding (QISD)

- **We implement and analyze quantum Gauss-Jordan elimination**

1. Randomly select as many columns as the number of rows in the public key

2. The matrix constructed in this way is an information set, if the information set is invertible? Perform Gaussian Elimination

3. We can compute the inverse matrix of the information set → Information set^{-1}

4. Check $C \times$ Information set^{-1} 's weight

Classical

Grover on ISD

1. Input Setting using butterfly network

2. Implement Quantum Gaussian Elimination

3. Weight check module for qubit vector

Quantum

Quantum Gauss-Jordan Elimination

- **Previous work uses Grover's algorithm to implement quantum Gauss-Jordan elimination --> High quantum cost**

QUANTUM GAUSS JORDAN ELIMINATION

DO NGOC DIEP[1] AND DO HOANG GIANG[2]

.

.

.

2. QGJE ALGORITHM

Step 1 Use the Grover's Search algorithm to find out the first non-zero $a_{i1} \neq 0$.
Step 2 If the search is successful, produce the first leading **1** in the first place as a_{11}, else change to the next column and repeat step 1.
Step 3 Eliminate all other entries $a_{1,1}, \ldots, a_{N,1}$ in the column.
Step 4 Change N to $N-1$, control if still $N > 0$, repeat the procedure from the step 1.
Step 5 In backward eliminate all $a_{N-1,N}, \ldots, a_{1,N}$.
Step 6 Check if $N > 0$, change N to $N-1$ and repeat the step 5.

D. N. Diep, D. H. Giang, "**Quantum Gauss Jordan Elimination**", International Journal of Theoretical Physics, Vol. 56, 2017.

Quantum Gauss-Jordan Elimination

- **We need to implement the following as a quantum circuit.**
  - **Swaps between rows**
  - **Eliminations between rows**

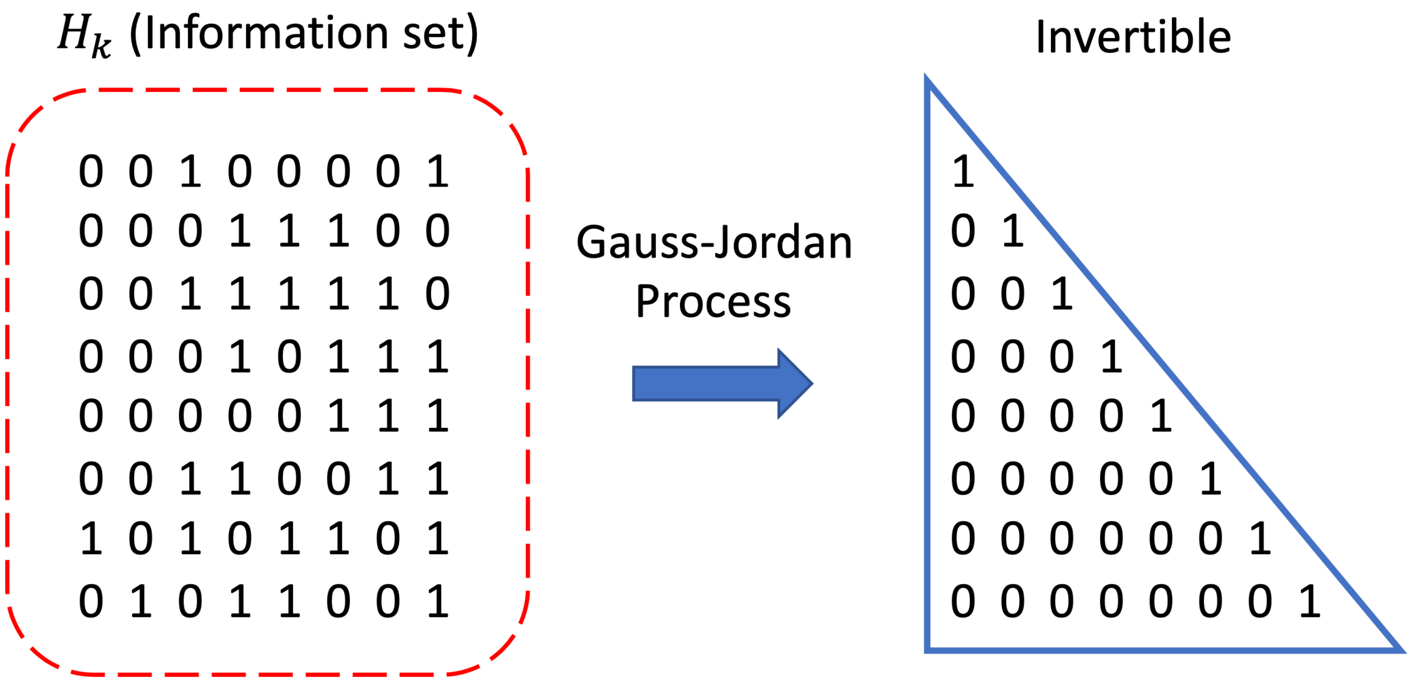


Fig. 4. Primary goal of Gauss-Jordan elimination.

Quantum Gauss-Jordan Elimination

- **We implement quantum Gauss-Jordan elimination only with quantum gates**
  - **Elimination → implemented with CCX, CCCX gates**
  - **Swaps → Implemented with Multi-Controlled Swap gates**

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

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

Implementation details can be found in the paper…

Quantum Gauss-Jordan Elimination

- **Estimation of quantum resources required for quantum Gaussian-Jordan elimination**

- 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
  → **It is more efficient than using Grover's algorithm**, **but requires a lot of cost.**

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

Analysis & Conclusion

- **Disadvantage of code-based ciphers**
  - **The key size is very large (very large memory capacity**
  - **Quantum attackers need very large memory (large qubits) and cost (gates and depth)to attack**

3.1 Parameter set kem/mceliece348864

KEM with $m = 12$, $n = 3488$, $t = 64$. Field polynomials $f(z) = z^{12} + z^3 + 1$

- **mceliece348864 public key size → (768 X 3488) parity check matrix**
  - **In the case of QISD, setting the public key is an unconditional option**
  - **2,678,784 (2.6 million) qubits required for public key setting**

- **Therefore, code-based cryptography is sufficiently resistant to quantum computers.**
  - **It is practically impossible to attack unless a new attack algorithm other than ISD comes out.**

Thank you!