

Optimized quantum circuit implementation of HQC core arithmetic

Sejin Lim

Hansung University

Contents

01. Introduction

02. Background

03. Quantum Implementation of HQC


04. Conclusion



01. Introduction

- The development acceleration of quantum computers hastens **the weakening and neutralizing period of existing cryptographic algorithms' security levels**
 - **By the quantum algorithm (Grover and Shor algorithm)**
- To prepare for this, NIST held a contest on Post Quantum Cryptography (PQC), which is safe even if quantum computers appear
 - Standardization of public key cryptography
- **By performing cryptographic analysis on a quantum computer, the security level of cryptographic algorithms can be checked**
 - For this purpose, cryptography must be implemented as a quantum circuit
 - We propose a quantum circuit implementation **optimized for binary field arithmetic**, which is **the core of key generation and encoding operation of HQC** (4th round candidate algorithm of PQC competition), and perform resource estimation

02. Background : Code-based Cryptography

- Code-based cryptography is based on **the syndrome decoding problem**, which is an NP-complete problem
- **Syndrome decoding problem**
 - 
 - $S = H e^T$ (Both S and H belong to Binary field)
Syndrome value (ciphertext)
 - The syndrome value S is generated by multiplying H and a secret vector e with a specific hamming weight (the number of non-zero bits in the code)
 - It is difficult to find out e even if you know H and S
- **Code-based Cryptography**
 - Intentionally injecting errors into messages, so that **only users who know the error can decrypt the message**
 - Since matrix operation is used, it has the advantage of fast encryption/decryption operation speed, but has the disadvantage of large key size

02. Background : HQC

HQC(Hamming Quasi-Cyclic)

- Code-based cryptographic schemes that utilizes the Hamming metric and random Quasi-Cyclic codes
- Quasi-Cyclic codes are designed to have a cyclic relationship among some of the matrix rows, enabling efficient computations
 - By exploiting this property, **only the first row needs to be stored, reducing the key size efficiently**
- The HQC paper presents **Public Key Encryption (PKE)** and **Key Encapsulation Mechanism (KEM)**
- Decoding is impossible because the error e added during encryption is very large, but only the user who has the secret key can easily decode by reducing e
- Decoding may fail with probability, but in the HQC paper, the authors demonstrate through a detailed and precise mathematical analysis that the probability of failure is negligibly low → Offer high security

02. Background : HQC

<HQC's PKE structure>

- **Key generation step** to generate public key and private key
 - $secret\ key = (x, y)$
 - $public\ key = (h, s)$
 - $s \leftarrow x + hy$
- **Encryption (Encoding) step** to generate ciphertext from the message
 - $ciphertext = (u, v)$
 - $v \leftarrow mG + sr_2 + e$
 - $u \leftarrow r_1 + hr_2$
- **Decryption (Decoding) step** to recover messages by removing errors from ciphertext
 - $Decode(v - u \cdot y)$

<In binary field arithmetic >

- Primitive polynomial $\mathbb{F}_2^n / \frac{X^n - 1}{(X - 1)}$ is used. (where n must be a prime number)
- For hqc-128, n is set to 17669
 - Based on the factorization property of the preceding expression, $\mathbb{F}_{2^{17668}} / (X^{17668} + X^{17667} + \dots + X + 1)$ is used

03. Quantum implementation of HQC

- Core Arithmetic in PKE

- Binary Field arithmetic ($\mathbb{F}_{2^{17668}}$, $\mathbb{F}_{2^{35850}}$, $\mathbb{F}_{2^{57636}}$)
 - Addition, Multiplication
- Syndrome computation
 - Binary Field arithmetic (Addition, Multiplication)
 - Matrix and Vector Multiplication
- Error correction (Future work)
 - Reed-Muller and Reed-Solomon (RMRS) concatenated codes

} Key Gen

} Encryption

} Decryption

03. Quantum implementation of HQC : Key Gen

- Core Operation in Key Gen

public key = (h, s) secret key = (x, y)

- **binary field arithmetic** : Depending on the security level, use $\mathbb{F}_{2^{17668}}[X]$, $\mathbb{F}_{2^{35850}}[X]$, $\mathbb{F}_{2^{57636}}[X]$
 - Use an irreducible polynomial of $(x^{n-1} + \dots + x + 1)$
 - $\mathbb{F}_{2^{17668}}/(x^{17668} + x^{17667} + \dots + x + 1)$
 - Security level 1 **simulation impossible** due to **large field size**
 - Implemented by reducing the field size
 - $\mathbb{F}_{2^{12}}/(x^{12} + x^{11} + \dots + x + 1)$
- $s \leftarrow x + hy$ (x, y, h existing on the same binary field)
- Cost : Addition < **Multiplication**

HQC	Field size	Public key (h, s)	Secret key (x, y)
hqc-128 (Level-1)	17,668	2,249 bytes	56 bytes
hqc-192 (Level-3)	35,850	4,522 bytes	64 bytes
hqc-256 (Level-5)	57,636	7,245 bytes	72 bytes

03. Quantum implementation of HQC : Key Gen

- Cost : Addition < **Multiplication**
- Addition is a simple XOR operation → Simple implementation with **only CNOT gate**
- Multiplication is AND & XOR operations → implementation with **Toffoli gate**
 - Toffoli gate is implemented using the T gates (**high cost**)
 - We have implemented the Toffoli gate with a T-depth of 2 for each instance
- **Multiplication** implementation : Application of [WISA'22] technique
 - Recursively apply Karatsuba Algorithm
 - **Optimize Toffoli-depth to 1 regardless of field size** by allocating ancilla qubits
→ **Can be multiplied by forming the overall depth very small**

Binary Field	Arithmetic	Qubits	Clifford	T gates	T-depth	Full depth
$\mathbb{F}_{2^{12}}$	Addition	24	12	.	.	1
	Multiplication	162	927	378	4	39

< Quantum cost for HQC binary field arithmetic $\mathbb{F}_{2^{12}}$ >

03. Quantum implementation of HQC : Encryption

- **Encryption**

- $u \leftarrow r_1 + hr_2$ (r_1, r_2 existing on the same binary field)
- $v \leftarrow mG + sr_2 + e$
- Compute and return $c = (u, v)$

- **Core Operation in Encryption**

→ binary field arithmetic (same as Key Gen)

→ Syndrome (v) computation : Matrix (Generator) and Vector (Message) multiplication



The matrix of G is used as is, without any modifications which is a feature and advantage of HQC

03. Quantum implementation of HQC : Encryption

- **Classical – Quantum implementation (Naïve, out-of-place)**
 - Matrix G (Generator) is a classical state, only vector m (message) is a quantum state
 - After allocating the qubit vector for the result value, CNOT according to the bit value (1) of G
- **Encryption (Matrix \times Vector) resource comparison**
 - Targeted a 12 x 24 matrix.
 - It is scalable and the actual matrix is very large.
 - Below is the exceptional result for a small matrix (12 x 24)

Method	Qubits	CNOT	Toffoli	Full Depth
C-Q (Naïve, out-of-place)	36	78	.	19

03. Quantum implementation of HQC

- Quantum Circuit Implementation Cost for Key Gen & Encryption

STEP	Arithmetic	Qubits	Clifford gates	T gates	T-depth	Full depth
Key Generation	Addition, Multiplication	174	939	378	4	40
Encryption	Addition, Multiplication, (Matrix \times Vector) Multiplication	234	1968	756	8	72

< Core arithmetic operation quantum circuit implementation in $\mathbb{F}_{2^{12}}/(x^{12} + x^{11} + \dots + x + 1)$ >

- Binary field multiplication
 - Performed once during the key generation
 - Performed twice during the encryption
 - **The number and depth of T gates are doubled**
- Use multipliers from WISA'22 → **T-depth is very low even though many T gates are used**
- Optimizing multiplication operations on binary fields is key to reducing the cost of implementing HQC circuits**

04. Conclusion

- In this paper, we propose a quantum circuit implementation optimized for binary field arithmetic, which is the core in key generation and encoding operations of HQC, and perform resource estimation
- In particular, multiplication operations on binary fields are optimized by applying the latest implementation techniques to reduce quantum resource costs
- This paper is meaningful in that it is **the first research on the implementation of quantum circuits in HQC**
- In the future, we plan to complete the implementation of HQC's quantum circuit by implementing the decoding step, and expand the binary field to the maximum by adjusting the range that can be simulated

Thank you for listening 😊