Quantum Implementation of Core Operations in Classic McEliece

Yujin Oh, Kyungbae Jang, Sejin Lim, Yujin Yang, Hwajeong Seo

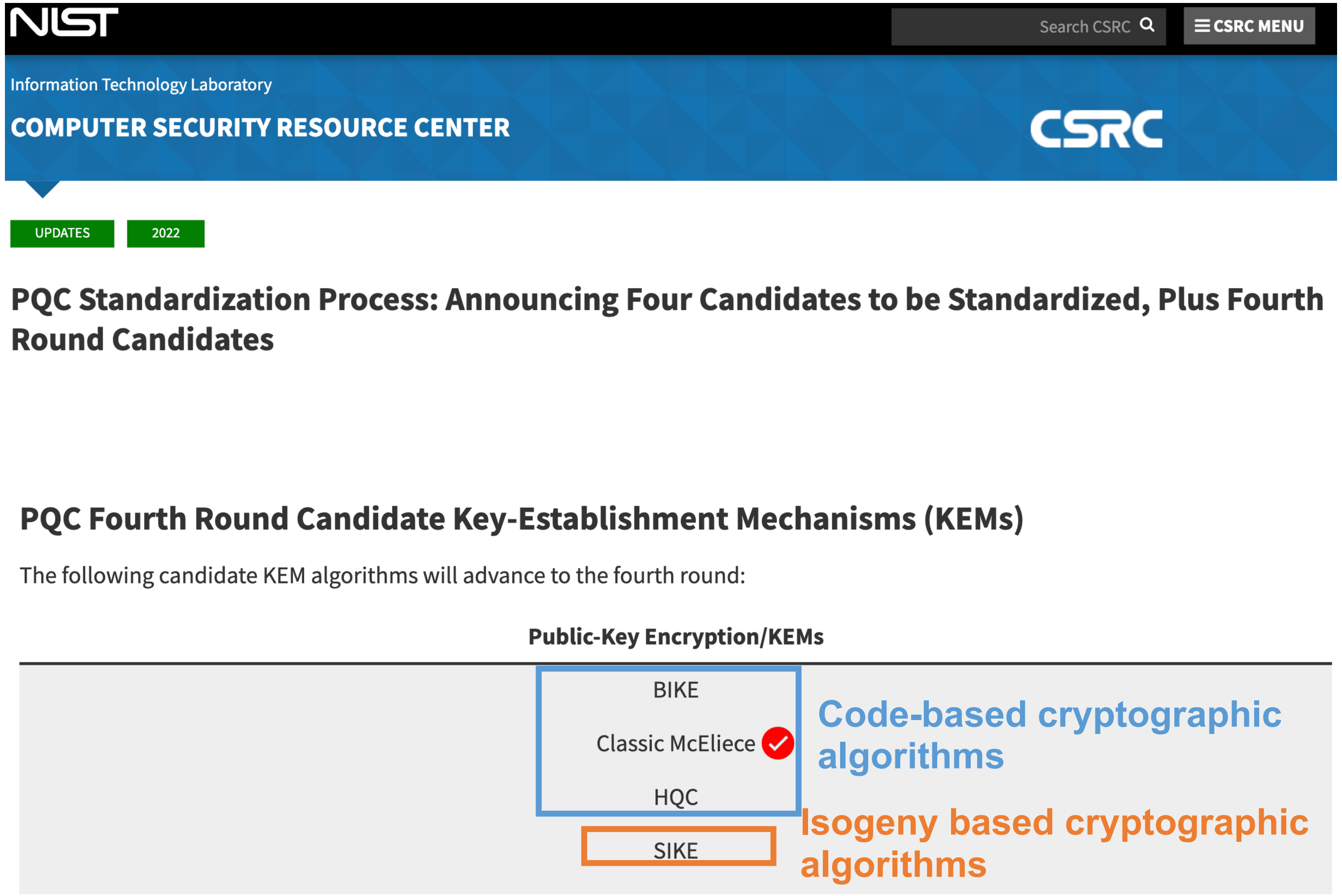
HSU 한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

- Introduction & Contribution
- Background
- Proposed Method
- Performance
- Conclusion

Introduction

- Post-quantum cryptography standardization contest in progress in NIST



- of quantum computers for cryptographic analysis undermines the security of existing encryption methods and diminishes their security strength.

- In order to establish post-quantum cryptographic systems, it becomes imperative to reevaluate the security of encryption algorithms in the context of quantum computing.

- In light of this, our paper focuses on optimizing the quantum circuit implementation technique for Classic McEliece, one of the candidate algorithms in NIST's Post-Quantum Cryptography Standardization Round 4.

Contribution

1. Quantum circuit of the encoding using three method.

→ We use three method to implement matrix-vector multiplication of the encoding in Classic McEliece and compare the resources of them

2. Quantum circuit of the Berlekamp-Massey algorithm

→ We present for the first time a quantum circuit of the Berlekamp-Massey decoding algorithm, the core operation of Classic McEliece

3. Efficient quantum implementation with WISA'22 quantum multiplication

→ We use the technique of [8] to implement binary field multiplication and inverse operation, which are key operations in the Berlekamp-Massey decoding algorithm. Through this, our proposed quantum circuit provides relatively low T -depth and full depth.

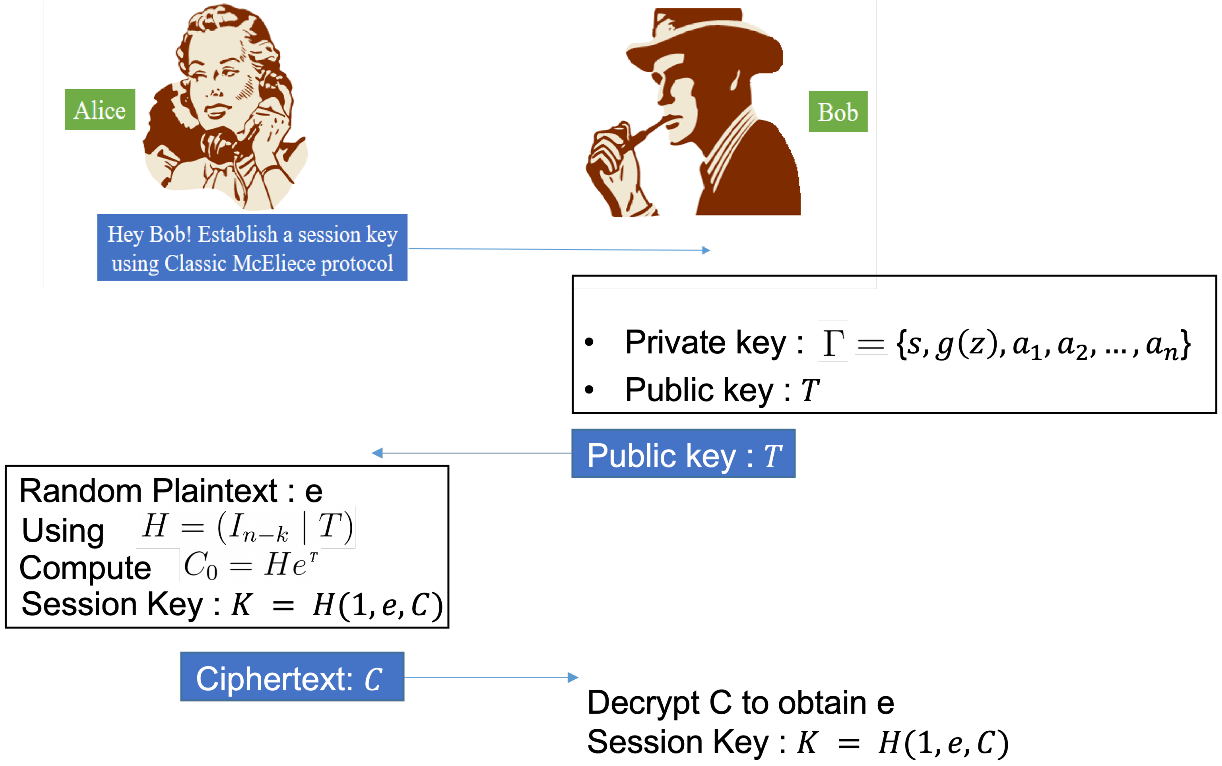
Background : Classic McEliece

- Classic McEliece

  - Classic McEliece is a code-based cipher which is one of the NIST Round 4 candidate algorithms.

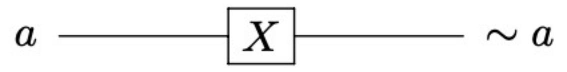
  - It uses the parity check matrix generated from the Goppa code as a public key.
    → A ciphertext is a syndrome value computed by multiplying a public key(a parity check matrix) and a secret information(a low-weight vector).

  - For decryption, the syndrome value is decoded using a private key and a decoder.
    → By performing syndrome decoding, a low-weight vector is recovered from the syndrome value.

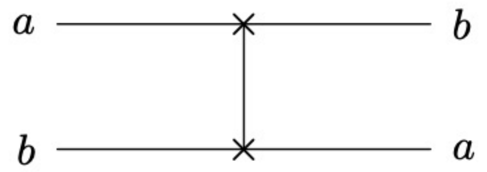


Background : Quantum gates

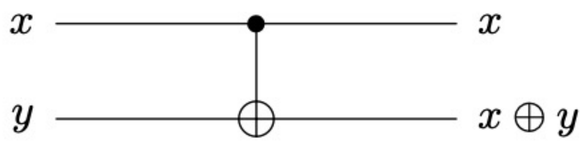
- Reversible quantum circuits for ciphers can be implemented using a variety of representa tive quantum gates.



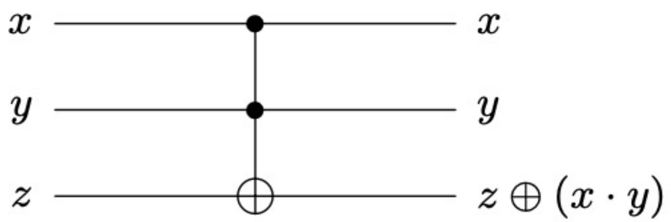
(a) X gate



(b) Swap gate



(c) CNOT gate



(d) Toffoli gate

Proposed Method

- Quantum Binary Field Multiplication
  - We apply **WISA'22 [8] multiplication**

    → optimized with a **Toffoli depth of one** for any field size.

  - WISA'22[8] multiplication
    - Using the **Karatsuba algorithm recursively** and allocating additional ancilla qubits.

      →All the AND operations become independent and the operations of **all Toffoli gates in parallel.**

      →The allocated ancilla qubits can be **reused** through **reverse operations**.

TABLE I: Quantum resources for the quantum multiplication circuit of $\mathbb{F}_{2^{12}}$ and $\mathbb{F}_{2^{13}}$.

Binary Field	#Clifford	#T	T-depth	#Qubits	Full depth
$\mathbb{F}_{2^{12}}$	761	378	4	162	37
$\mathbb{F}_{2^{13}}$	966	462	4	198	54

[8] K. Jang, W. Kim, S. Lim, Y. Kang, Y. Yang, and J. Hwa, "Optimized implementation of quantum binary field multiplication with toffoli depth one," in International Conference on Information Security Applications.

Proposed Method

- Quantum Binary Field Inversion
  - The inversion based on **Itoh-Tsujii**
  - Performing operations using **multiplications and squarings.**
    → Using the previously implemented multiplication and squaring
  - The inversion using **WISA'22[8] multiplication.**
    → Across multiple multiplication operations, **ancilla qubits can be reused.**
  - The inversion using **LUP decomposition for squarings.**
    → Through LUP decomposition, implemented **in-place** using only **CNOT gates.**

[8] K. Jang, W. Kim, S. Lim, Y. Kang, Y. Yang, and J. Hwa, "Optimized implementation of quantum binary field multiplication with toffoli depth one," in International Conference on Information Security Applications.

Proposed Method

- Quantum Binary Field Inversion
  - **Ancilla qubits** are allocated only in the first multiplication and subsequent multiplications **reuse** the previous ancilla qubits without additional ancilla qubits.

TABLE II: Quantum resources for the quantum inversion circuit of $\mathbb{F}_{2^{12}}$ and $\mathbb{F}_{2^{13}}$.

Binary Field	#Clifford	#T	T-depth	#Qubits	Full depth
$\mathbb{F}_{2^{12}}$	4758	1890	20	402	194
$\mathbb{F}_{2^{13}}$	4988	1848	16	422	369

Algorithm 1 Inversion quantum circuit of $\mathbb{F}_{2^{12}}$

Input: 12-qubit x, 12-qubits $temp_{0\sim6}$, ancilla qubits ac

Output: x^{-1}

$temp_0 \leftarrow$ CNOT32$(x, temp_0)$
$temp_0 \leftarrow$ Squaring$(temp_0)$
$temp_1 \leftarrow$ Multiplication$(x, temp_0, ac)$
$temp_2 \leftarrow$ CNOT32$(temp_1, temp_2)$
$temp_1 \leftarrow$ Squaring$(temp_1)$
$temp_1 \leftarrow$ Squaring$(temp_1)$
$temp_3 \leftarrow$ Multiplication$(temp_2, temp_3, ac)$
$temp_4 \leftarrow$ CNOT32$(temp_3, temp_4)$
$temp_3 \leftarrow$ Squaring$(temp_3)$
$temp_3 \leftarrow$ Squaring$(temp_3)$
$temp_3 \leftarrow$ Squaring$(temp_3)$
$temp_3 \leftarrow$ Squaring$(temp_3)$
$temp_5 \leftarrow$ Multiplication$(temp_3, temp_4, ac)$
$temp_5 \leftarrow$ Squaring$(temp_5)$
$temp_5 \leftarrow$ Squaring$(temp_5)$
$temp_6 \leftarrow$ Multiplication$(temp_2, temp_5, ac)$
$temp_6 \leftarrow$ Squaring$(temp_6)$
$temp_7 \leftarrow$ Multiplication$(x, temp_6, ac)$
$temp_7 \leftarrow$ Squaring$(temp_7)$
return $temp_7$

Proposed Method

- The encoding performs **matrix-vector multiplication**. (public key H and the random vector e)

  $C_0 = He$ (The public key H is a pre-defined value).

  - Quantum simulation of the smallest parameter public key (768 x 3844) matrix is not possible.

    → Performing **reduced matrix-vector** multiplication (8 x 16)

  - We implement three methods and compares their circuit costs.

    - Quantum – Quantum operation
    - Naïve (Out-of-place)
    - LUP decomposition(in-place)

Proposed Method

- Quantum – Quantum
  - When both qubits are set to 1, add 1 to the result vector.
  - Perform **AND** operations between the matrix values and vector qubits, and then implement the **XOR** operation with the result vectors.
    → **Using Toffoli gates**.

$$\begin{bmatrix} 0&0&0&0&0&1&0&1 \\ 0&0&1&0&1&0&1&0 \\ 1&0&0&0&0&1&1&0 \\ 1&0&1&1&1&0&0&0 \\ 0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&0&1 \\ 1&1&1&1&1&1&0&1 \\ 1&1&0&0&1&1&1&0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

```
def Encoding_1(eng, h, e, col, row):

    syndrome = eng.allocate_qureg(row)

    for i in range(row):
        # Quantum - Quantum
        h_e_mul(eng, h[(col*i):(col*i)+col], e, syndrome[row-1-i], col)

    return syndrome
```

Proposed Method

- Classical - Quantum (Naïve)
  - Depending on the value of H, perform a **XOR operation**(CNOT gate) on the result vector. **(out-of-place)**
    → When the value of H is 1 , perform the CNOT operation between the the vector e and re sult qubit.

$$\begin{bmatrix} 0&0&0&0&0&1&0&1 \\ 0&0&1&0&1&0&1&0 \\ 1&0&0&0&0&1&1&0 \\ 1&0&1&1&1&0&0&0 \\ 0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&0&1 \\ 1&1&1&1&1&1&0&1 \\ 1&1&0&0&1&1&1&0 \end{bmatrix} \times \begin{bmatrix} 1\\1\\0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 0\\0\\1\\1\\0\\0\\0\\0 \end{bmatrix}$$

```
def Encoding_2(eng, H, e, col, row):

    syndrome = eng.allocate_qureg(row)
    for i in range(row):
        for j in range(col):
            # Classic - Quantum
            if(H[col*i+j] == 1):
                CNOT | (e[col-1-j], syndrome[row-1-i])
```

Proposed Method

- Classical - Quantum (LUP decomposition)
  - **In-place** implementation based on the **LUP decomposition** of matrix H
    → the result vector is computed directly.
  - Through LUP decomposition , we obtain three matrices :
    (a permutation matrix, a lower triangular matrix, and an upper triangular matrix)
    → **We can achieve the implementation using only CNOT gates without ancilla qubits to s tore the result.**

LUP decomposition

$$\begin{bmatrix} 0&0&0&0&0&1&0&1\\ 0&0&1&0&1&0&1&0\\ 1&0&0&0&0&1&1&0\\ 1&0&1&1&1&0&0&0\\ 0&0&0&1&1&1&0&1\\ 0&0&1&0&1&1&0&1\\ 1&1&1&1&1&1&0&1\\ 1&1&0&0&1&1&1&0 \end{bmatrix} \times \begin{bmatrix} 1\\1\\0\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 0\\0\\1\\1\\0\\0\\0\\0 \end{bmatrix}$$

```
def Apply_U(eng, vector_e, row, col, U):
    for i in range(row):
        for j in range(col - 1 - i):
            if (U[(i * col) + 1 + i + j] == 1):
                CNOT | (vector_e[col - 2 - i - j], vector_e[col - 1 - i])
```

Proposed Method

- Quantum Circuit Implementation of the Berlekamp-Massey Decoding Algorithm

  - In Classic McEliece, the BM algorithm recovers the vector of secret weight–t from the ciphertext syndrome value.

  - The fundamental operations involve arithmetic in $\mathbb{F}_{2^{12}}/(x^{12}+x^3+1)$, $\mathbb{F}_{2^{13}}/(x^{13}+x^4+x^3+\ x^1+1)$

  - Multiplication and inversion are repeated as key operations.
    → **WISA'22[8] multiplication**
    → **Inversion based on Itoh-Tsujii**

  - We only target the Berlekamp-Massey decoding algorithm in **mceliece348864.**

[8] K. Jang, W. Kim, S. Lim, Y. Kang, Y. Yang, and J. Hwa, "Optimized implementation of quantum binary field multiplication with toffoli depth one," in International Conference on Information Security Applications.

Proposed Method

- Quantum Circuit Implementation of the Berlekamp-Massey Decoding Algorithm

  - In the implementation of Berlekamp-Massey, we employ the wisa multiplication technique. By using this multiplication approach, we can **reuse the ancilla qubits**.

  - The application of this multiplication technique allows us to reduce the number of qubits and consequently enables the reuse of **the initially allocated ancilla qubits** across multiple multiplication and inversion operations.

Algorithm 2 The Berlekamp-Massey quantum circuit of Classic McEliece

Input: 12-qubit b, 12-qubit array $T[t+1]$, $C[t+1]$, $B[t+1]$, $s[2t]$, ancilla qubits ac, $L = 0$ (classical)
Output: C

```
b ← X(b[0])
C[0] ← X(C[0][0])
B[1] ← X(B[1][0])
for N = 0 to 2t − 1 do
    d ← new 12-qubit allocation
    for i = 0 to min(N, t) do
        d ← MultiplicationXOR(C[i], s[N − i], ac)
    end for
    if (2L ≤ N) then
        for i = 0 to t do
            T[i] ← new 12-qubit allocation
            T[i] ← CNOT32(C[i], T[i])
        end for
    end if
    b^−1 ← Inversion(b, ac)
    if (2L > N) then
        for i = 0 to t do
            C[i] ← MultiplicationXOR(f, B[i], ac)
        end for
    end if
    if (2L ≤ N) then
        for i = 0 to t do
            C[i] ← MultiplicationXOR(f, B[i], ac)
            L ← N + 1 − L (classical)
        end for
        for i = 0 to t do
            B[i] ← T[i]
        end for
        b = d (classical)
    end if
    for i = 0 to t − 1 do
        B[t − i] ← B[t − 1 − i]
    end for
    B[0] ← new 12-qubit allocation
end for
return C
```

Performance

- Estimating the quantum resources for the quantum circuit of the matrix-vecto r multiplication encoding algorithm.
  - The **LUP decomposition** allows for an **in-place** implementation, **minimizing the requi red number of qubits.**
  - In terms of CNOT gate count and **depth**, both naïve and LUP offer similar performance.
  - Quantum– Quantum method achieves a **highest cost** in terms of both depth and qubits.

TABLE III: Quantum resources for the quantum circuit of the matrix-vector multiplication encoding algorithm.

matrix-vector encoding	Method	#Clifford	$\#T$	T-depth	#Qubits	Full depth
Quantum-Quantum	Naive	784	896	92	152	147
Classic-Quantum	Naive	45	-	-	24	14
	LUP	37	-	-	16	13

Performance

- Estimating the quantum resources for the quantum circuit of the Berlekamp-Massey decoding algorithm
  - **Multiplication and inversion** are operations that require many quantum resources and are repeated in the proposed **Berlekamp-Massey** quantum circuit.
    → many quantum resources are used and the number of **qubits** is also very **high**.
    → Because the parameters used in Classic McEliece are large, this has a high cost even when ported to quantum.

TABLE IV: Quantum resources for the quantum circuit of the Berlekamp-Massey decoding algorithm.

Berlekamp-Massey	#Clifford	#T	T-depth	#Qubits	Full depth
$t = 64$ and $\mathbb{F}_{2^{12}}$	12823392	579384	60800	888492	363696

Conclusion

- In this paper, the core operations of the Classic McEliece algorithm, which is one of the code-bas ed cryptosystems among the NIST Round4 candidates, are implemented as quantum circuits.
- Especially, the quantum circuit for **Berlekamp-Massey decoding** is presented for the first time in this work.
- As a future research direction, we plan to **complete the entire quantum circuit** for Classic McEli ece based on the implemented core operations.
- Given the significantly large key size of the Classic McEliece scheme, adjustments need to be ma de to accommodate the feasible simulation range.

Q & A