Classic McEliece 공개키 생성 양자회로 구현

오유진 * 장경배 * 임세진 * 서화정 *†
* 한성대학교 대학원 융합보안학과

서 론

● 양자 컴퓨터의 발전으로 Shor알고리즘의 공개키 암호 공격이 다항시간 내에 가능함이 밝혀지면서 NIST에서는 양자 내성 암호 공모전을 진행하였다. 현재 4라운드까지 진행되었으며 C)을 진행하였다. 현재 4라운드까지 진행되었으며 3개의 코드기반 암호(Classic McEliece, BIKE, HQC)와 아이소제니 암호(SIKE)가 후보로 선정되었다.
본 논문에서는 NIST의 양자내성암호 공모전 Round 4의 후보 알고리즘 중 하나인 Classic McEliece [3]에 대한 공개 키 생성 양자회로를 최적화하여 구현하며 회로에 필요한 양자 자원을 추정한다.

관련 연구

● Classic McEliece는 NIST Round 4 후보 알고리즘 중 하나인 코드기반 암호이다. Classic McEliece는 McEliece의 Goppa code와 Niderreiter의 패리티 체크 행렬을 사용한다. Goppa code에서 생성된 패리티 체크 행렬을 공개키로 사용한다. 10가지의 다른 파라미터가 있으며, mceliece348864의 경우에만 $\mathbb{F}_{2^{12}}/(x^{12}+x^3+1)$상에서 연산이 사용되고 이외의 파라미터들의 경우 $\mathbb{F}_{2^{13}}/(x^{13}+x^4+x^3+x+1)$ 상에서 연산이 사용된다.

본론

● 양자 시뮬레이션이 불가능함에 따라 본 구현은 **축소된 파라미터**(3488 x 64 → **32 x 4**)를 사용함
● 바이너리 필드 상에서의 덧셈, 곱셈, 역치 연산을 사용

● 역치 연산 내 제곱연산
- **LUP 분해** : CNOT 게이트와 Swap 게이트만을 사용하여 in-place로 구현

● 곱셈 연산
- Jang et al. 의 **WISA'22 곱셈기** – 카라추바 곱셈을 재귀적으로 적용하고 추가 큐비트를 할당하여 모든 곱셈을 병렬로 수행함으로써 모든 곱셈의 수행단위를 1로 줄여 필드에 상관없이 **Toffoli depth를 1로 최적화**
- 사용되는 추가 큐비트들은 **역연산**을 통해 재사용하여 **큐비트 수 최적화**

● 역치 연산
- 제곱과 곱셈 연산들의 조합으로 이루어진 **Itoh-Tsujii 알고리즘** 활용
- **LUP 분해**를 사용한 제곱연산과 Jang et al.**의 WISA' 22 곱셈기** 사용
- 반복되는 곱셈과 제곱 연산들로 인해 다른 산술들에 비해 높은 큐비트 수, 게이트 수와 depth를 요구

● 바이너리 필드 산술 양자 자원 추정 비용

Field	Arithmetic	Method	Qubits	Clifford gates	T gates	T-depth	Full depth
$\mathbb{F}_{2^{12}}$	Addition	-	24	12	-	-	1
	Squaring	LUP	12	7	-	-	2
	Multiplication	Schoolbook	36	921	1008	136	307
		Jang et al.	162	761	378	4	37
	Inversion	Itoh-Tsujii	402	4758	1890	20	194
$\mathbb{F}_{2^{13}}$	Addition	-	26	13	-	-	1
	Squaring	LUP	13	23	-	-	14
	Multiplication	Schoolbook	42	1110	1183	148	333
		Jang et al.	198	966	462	4	54
	Inversion	Itoh-Tsujii	422	4988	1848	16	369

● Classic McEliece 공개키 생성 양자 회로 자원 추정 비용

Field	Qubits	Clifford gates	T gates	T-depth	Full depth
$\mathbb{F}_{2^{12}}$	68704	571712	270144	7040	12718
$\mathbb{F}_{2^{13}}$	79403	672448	309904	7360	18783

Algorithm 1 : pk_gen
Input: *GFBITS*-qubit b, *GFBITS*-qubit array, $G[T+1]$, $L[N]$, $\boldsymbol{inv}[N]$, $mat[T\times GFBITS][N/8]$, ac
Output: mat

```
for i = 0 to N
    inv[i] ← CNOT_gate(G[T], inv[i])
    for j = T−1 to 0
        inv[i] ← Multiplication(L[i], inv[i], ac)
        inv[i] ← CNOT_gate(G[j], inv[i])
for i = 0 to N
    inv[i] ← Inversion(inv[i], ac)
for i = 0 to T
    for j = 0 to N
        for k = 0 to GFBITS
            b ← new GFBITS -qubit allocation
            b ← CNOT_gate(inv[j+7], b)
            b ← RightShift(b, k)
            b ← LeftShift_one(b)
            for l = 6 to 0
                ancilla ← new GFBITS-qubit allocation
                ancilla ← CNOT_gate(inv[j+l], ancilla)
                ancilla ← RightShift(ancilla, k)
                b ← OR_gate(b, ancilla)
                if(i ≠ 0) :
                    b ← LeftShift_one(b)
            mat[i • GFBITS+k][j/8] ← b
            j = j+8
for j = 0 to N
    inv[j] ← Multiplication(L[j], inv[j], ac)
: return mat
```

축소된 파라미터를 사용함에도 불구하고 높은 비용이 요구되는 양자 곱셈과 역 연산들이 다수 반복됨에 따라 매우 높은 비용의 양자 자원이 사용된다.

결론

● 양자 후 보안 강도를 위해, 양자 회로를 구현하고 자원들을 추정하는 것은 중요하다. 본 논문에서는 NIST Round4 후보 알고리즘인 Classic McEliece 공개키 생성 양자 회로를 구현하고 이에 필요한 양자 자원들을 추정하였다. 사용되는 필드 산술들에 대한 최적화를 통해 현재 시뮬레이션 문제로 정확한 파라미터들에 대한 구현 및 추정하는 것은 어려우며 그로 인해 축소된 파라미터를 사용하였다. 제안된 구현은 파라미터 값을 조정함으로써 확장이 가능하다. 이는 향후 연구에서 양자 시뮬레이션 가능 범위를 크게 조정하는 방향으로 발전할 것 이다.