Classic McEliece 공개키 생성 양자회로 구현

오유진[1], 장경배[2], 임세진[3], 서화정[4]
[1]한성대학교 융합보안학과 석사과정
[2]한성대학교 정보컴퓨터공학과 박사과정
[3]한성대학교 IT융합공학부 석사과정
[4]한성대학교 융합보안학과 교수
oyj0922@gmail.com, starj1023@gmail.com, dlatpwls834@gmail.com,
hwajeong84@gmail.com

Implementation of Classic McEliece Public Key Generation Quantum Circuit

Yu-jin Oh[1], Kyung-bae Jang[2], Se-jin Lim[3], Hwa-jeong Seo[4]
[1,4]Dept. of Convergence Security, Han-Sung University
[2]Dept. of Computer Information Engineering, Han-Sung University
[3]Dept. of It Convergence Engineering, Han-sung University

요 약

양자 알고리즘인 Shor 알고리즘으로 공개키 암호의 보안성이 붕괴됨에 따라 NIST는 양자내성암호 표준화 공모전을 진행하였다. 또한 암호시스템의 양자 후 보안을 위해, 양자 컴퓨터상에서 암호 알고리즘들을 미리 구현하고 분석하는 연구가 진행되고 있다. 이에 본 논문에서는, NIST 양자내성암호 공모전 Round 4의 후보 알고리즘인 Classic McEliece의 공개키 생성 양자 회로 구현을 제시하고 회로에 필요한 양자 자원을 추정한다.

1. 서론

양자 컴퓨터의 발전으로 현재 암호 체계가 위협받고 있는 상황이다. 양자 알고리즘인 Shor 알고리즘 [1]으로 인해 공개키 암호의 보안성이 붕괴됨이 밝혀짐에 따라 NIST에서는 양자 컴퓨터에 내성을 갖는 양자내성암호 표준화 공모전(PQC)을 진행하였다. 현재 4라운드까지 진행되었으며 3개의 코드기반 암호(Classic McEliece, BIKE, HQC)와 아이소제니 암호(SIKE)가 후보로 선정되었다. 또한 Grover 알고리즘 [2]으로 대칭 키 암호의 공격 비용을 루트만큼 ($O(N) \rightarrow O(\sqrt{N})$) 감소시킬 수 있다. 이에 안전한 양자 후 보안성을 위해 암호 알고리즘들을 양자 컴퓨터 상에서 분석하는 연구들이 진행되고 있다.

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.

2. 관련연구

2-1. Classic McEliece

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.

3. Classic McEliece 공개키 생성 양자회로 구현

제안하는 Classic McEliece 공개키 생성 양자 회로는 (그림 1)과 같으며, 가장 작은 파라미터인 mceliece348864에 대한 양자 시뮬레이션이 불가능함에 따라 본 구현 및 비용 분석에서는 축소된 파라미터(3488 x 64 → 32 x 4)를 사용하며 필드 사이즈는 동일하게 $\mathbb{F}_{2^{12}}$, $\mathbb{F}_{2^{13}}$ 을 사용한다.

Classic McEliece의 키 생성에는 바이너리 필드 산술이 사용되며 덧셈, 곱셈, 역치 연산을 핵심으로

구성된다.

바이너리 필드 상에서의 덧셈 연산은 XOR 연산과 동일하므로 필드 크기만큼 CNOT게이트를 사용하여 구현할 수 있으며 병렬 연산으로 인해 depth 1로 구현된다.

역치 연산에서 사용되는 제곱 연산은 선형 연산으로 결과 값을 모듈러로 줄이는 과정 (XOR 연산)으로 구현된다. 기본적인 방법으로는, CNOT 게이트만을 사용하여 구현하거나, LUP 분해를 기반으로 CNOT, Swap 게이트를 사용하여 in-place로 구현할 수 있다. 이 경우, 결과 값을 저장할 보조 큐비트 할당이 필요하지 않으며 Swap 게이트는 큐비트 배열의 인덱스를 변경하는 logical Swap으로 구현이 가능하기 때문에 비용을 차지하지 않는다. 이러한 이점으로, 제곱 연산 시, LUP분해 기법을 사용하여 큐비트 수를 줄이고 오직 CNOT 게이트만을 이용하여 depth 측면에서도 최적화를 진행한다.

바이너리 필드 곱셈 연산의 경우, 높은 연산 비용이 요구되며 곱셈 연산 후 필드 사이즈에 따른 모듈러 축소를 통해 연산된다. 일반적인 곱셈 연산으로는 Schoolbook 곱셈과 카라츄바 곱셈이 있으며 Schoolbook의 경우 n^2, 카라추바는 $n^{\log_2 3}$의 AND 연산이 필요하다. 양자 컴퓨터 상에서의 AND 연산은 Toffoli 게이트 연산과 동일하며 Toffoli 게이트는 양자 게이트 중 높은 비용을 요구하기 때문에 이를 최적화 하는 것이 중요하다. [4]에서는 기본적인 Schoolbook 양자 곱셈을 구현하였다. [5]의 기법에서는 최적화를 위해 카라추바 곱셈을 재귀적으로 적용하여 모든 곱셈의 수행 단위를 1로 줄여 Toffoli depth를 감소시킨다. 또한 추가 큐비트를 할당하여 곱셈을 병렬로 수행함으로써 필드 크기에 상관없이 Toffoli depth가 1로 최적화하여 depth 측면에서 최적화하였다. 본 논문에서는 depth를 중심으로 [5]의 기법을 적용하여 Toffoli depth와 Full depth 측면에서 최적화 한다. 또한 이 기법을 사용함으로써 반복되는 곱셈 연산에서 사용되는 보조 큐비트를 재사용하여 큐비트 측면에서도 최적화를 진행한다.

역치 연산 시, [6]의 기법을 적용한다. [6]의 기법은 제곱과 곱셈 연산들의 조합으로 이루어진 Itoh-Tsujii 알고리즘을 활용하여 구현된다. 또한 역치 연산 내부의 곱셈 연산 시에는 [5]의 기법을 사용하여 Toffoli depth를 1로 줄이고 보조 큐비트를 재사용할 수 있다. 이 역치 연산 기법을 적용함으로써 다수 반복되는 역치연산 및 곱셈 연산에 보조 큐비트를 추가로 할당없이 재사용할 수 있으며 Toffoli depth 및 Full depth 측면에서 최적화한다.

그러나 다양한 최적화 기법을 사용함에도 불구하고 여러번의 반복문이 실행되기 때문에 높은 depth와 큐비트가 사용된다.

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*

```
1: for i = 0 to N
2:    inv[i] ← CNOT_gate(G[T], inv[i])
3:    for j = T−1 to 0
4:       inv[i]← Multiplication(L[i],inv[i],ac)
5:       inv[i]← CNOT_gate(G[j],inv[i])
6: for i = 0 to N
7:    inv[i] ← Inversion(inv[i],ac)
8: for i = 0 to T
9:    for j = 0 to N
10:      for k = 0 to GFBITS
11:         b ← new GFBITS -qubit allocation
12:         b ← CNOT_gate(inv[j+7],b)
13:         b ← RightShift(b,k)
14:         b ← LeftShift_one(b)
15:         for l = 6 to 0
16:            ancilla ← new GFBITS-qubit allocation
17:            ancilla ← CNOT_gate(inv[j+l],ancilla)
18:            ancilla ← RightShift(ancilla,k)
19:            b ← OR_gate(b,ancilla)
20:            if(i ≠ 0) :
21:               b ← LeftShift_one(b)
22:         mat[i • GFBITS+k][j/8] ← b
23          j = j+8
23:   for j = 0 to N
24:      inv[j]← Multiplication(L[j],inv[j],ac)
25 : return mat
```

(그림 1) Classic McEliece 공개 키 생성 양자 회로 구현

3. 성능 평가 및 분석

본 장에서는 Classic McEliece 공개키 생성 양자 회로 구현에 필요한 양자 자원들을 추정한다. 자원 측정 시 Toffoli gate는 8개의 Clifford 와 7개의 T 게이트로 분해하여 측정하며 이에 따른 T-depth는 4이다.

Classic McEliece의 필드인 $\mathbb{F}_{2^{12}}$, $\mathbb{F}_{2^{13}}$ 상에서의 산술연산에 대한 자원 측정은 <표 1>과 같다. 제곱 연산의 경우, in-place 연산으로 필드 크기만큼만 큐비트를 사용한다. $\mathbb{F}_{2^{13}}$에 비해 $\mathbb{F}_{2^{12}}$ 에서는 모듈러 연산 시 더 적은 CNOT 게이트를 사용하며 많은 병렬 연산이 가능하기 때문에 depth 또한 2로 낮다. 곱셈 연산의 경우 [5]의 기법과 [4]에서 제시한

Schoolbook 양자 곱셈의 양자 자원 비용을 비교하여 볼 수 있다. [5]의 기법은 [4]와 비교했을 때 더 많은 큐비트를 사용하지만, 게이트 및 depth 측면에서 효과적으로 최적화된 것을 확인할 수 있다. 역치 연산의 자원 추정 비용을 보면, 반복되는 곱셈과 제곱 연산들로 인해 다른 산술들에 비해 높은 큐비트 수, 게이트 수와 depth를 요구한다.

<표 1> 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	-	12	7	-	-	2
	Multiplication	[3]	36	921	1,008	136	307
		[2]	162	761	378	4	37
	Inversion	[4]	402	4,758	1,890	20	194
$\mathbb{F}_{2^{13}}$	Addition	-	26	13	-	-	1
	Squaring	-	13	23	-	-	14
	Multiplication	[3]	42	1,110	1,183	148	333
		[2]	198	966	462	4	54
	Inversion	[4]	422	4,988	1,848	16	369

Classic McEliece 공개 키 생성 양자 회로 구현에 대한 자원 추정 결과는 <표 2>와 같다. 축소된 파라미터를 사용함에도 불구하고 높은 비용이 요구되는 양자 곱셈과 역치 연산들이 다수 반복됨에 따라 매우 높은 비용의 양자 자원이 사용된다.

<표 2> Classic Mceliece 공개 키 생성 양자 회로 양자 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

3. 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.

4. Acknowledgment

This work was 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, 80%) and this work was 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, 20%).

참고문헌

[1] Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. 1999, 41, 303-332.

[2] Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22-24 May 1996; pp. 212-219.

[3] Martin Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Nieder- hagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece, 2020.

[4] D. Cheung, D. Maslov, J. Mathew, D. K. Pradhan, On the design and optimization of a quantum polynomial time attack on elliptic curve cryptography, Theory of Quantum Computation, Communication, and Cryptography. 2007.

[5] K. Jang, W. Kim, S. Lim, Y. Kang, Y. Yan

and H. Seo, Optimized Implementation of Quantum Binary Field Multiplication with Toffoli Depth One, International Conference on Information Security Applications, 2022.
[6] K. Jang, W. Kim, S. Lim, Y. Kang, Y. Yan and H. Seo, Quantum Binary Field Multiplication with Optimized Toffoli Depth and Extension to Quantum Inversion, Sensors, 2023.