Classic McEliece 양자 회로 구현

오유진* 장경배* 임세진* 양유진* 서화정**

*한성대학교 (대학원생)

**한성대학교 (교수)

Quantum Circuit Implementation of Classic McEliece

Yu-jin Oh* Kyung-bae Jang* Se-jin Lim* Yu-jin Yang* Hwa-jeong Seo**

*Hansung University (Graduate student)

**Hansung University (Professor)

요 약

양자 알고리즘을 동작시킬 수 있는 대규모의 양자 컴퓨터가 개발된다면, 강력한 양자 암호 분석이 가능할 것으로 기대된다. 안전한 양자 후 암호 시스템 구축을 위해서, 암호 알고리즘들에 대한 양자 후 보안성 평가가 필요한 상황이다. 이에 다양한 양자 암호 분석 연구들이 다수 발표되고 있으며, 암호를 분석하기 위한 양자 회로의 자원을 최소화하는 구현 기법들이 제시되고 있다. 본 논문에서는, NIST 양자내성암호 공모전 Round 4의 후보 알고리즘인 Classic McEliece의 효율적인 양자 회로 구현을 제시한다. Classic McEliece의 키 생성, 인코딩, 디코딩 연산을 양자 회로상에서 최적화 구현하며, 구체적으로는 바이너리 필드 산술, 선형 연산, Berlekamp-Massey 디코딩 양자 회로를 그 대상으로 한다.

I. 서론

양자 컴퓨터는 특정 난제들을 효율적으로 모델링하고 빠르게 해결할 수 있다. 이러한 양자 컴퓨터를 사용한 암호 분석 기술은 기존 암호들의 보안성을 부정하거나 보안 강도를 감소시킨다. 안전한 양자 후 암호 시스템을 위해서는 기존 암호 알고리즘들의 보안성을 양자 컴퓨터상에서 재 분석해야함에 따라, 다양한 양자 암호 분석 연구들이 발표되고 있다. 이에 본 논문에서는 NIST의 양자내성암호 공모전 Round 4의 후보 알고리즘 중 하나인 Classic McEliece [1]에 대한 양자회로를 최적화하여 구현한다. 코드기반암호인 Classic McEliece의 양자 암호 분석을 위해, Goppa 코드에서 사용되는 바이너리 필드 산술, 인코딩, 디코딩 양자 회로를 제시한다.

II. Classic McEliece 양자 회로 구현

본 장에서는 Classic McEliece의 핵심 연산인 키 생성, 인코딩, 디코딩의 최적화된 양자 회로 구현 기법들에 대해 설명한다. [표 1~3]은 Classic McEliece에서 다루는 키 생성의 바이너리 필드의 산술, 인코딩의 행렬-벡터 곱셈, 마지막으로 디코딩의 Berlekamp-Massey 알고리즘을 양자 회로 상에서 최적화하여 구현한 결과를 포괄적으로 보여준다.

2.1 키 생성 양자 회로 구현

Classic McEliece의 키 생성에는 바이너리 필드 산술이 사용된다. mceliece348864의 경우 $\mathbb{F}_{2^{12}}/(x^{12}+x^3+1)$의 산술들이 사용되며, 이외의 파라미터들의 경우 $\mathbb{F}_{2^{13}}/(x^{13}+x^4+x^3+x+1)$의 산술이 사용된다. 덧셈 연산은 XOR 연산만이 사용되기 때문에, 필드 크기만큼의 양자 CNOT 게이트를 사용하여 depth 1로 구현된다.

제곱 연산은 제곱된 결과 값에 대한 모듈러 감소 (XOR 연산)만을 CNOT 게이트를 사용하여

Field	Arithmetic	Method	Qubits	Clifford gates	T gates	T-depth	Full depth
$\mathbb{F}_{2^{12}}$	Addition	-	24	12	-	-	1
	Squaring	-	12		-	-	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	7	-	-	2
	Multiplication	[3]	42	1,110	1,183	148	333
		[2]	198	966	462	4	54
	Inversion	[4]	422	4,988	1,848	16	369

[표 1] $\mathbb{F}_{2^{12}}/(x^{12}+x^3+1)$그리고 $\mathbb{F}_{2^{13}}/(x^{13}+x^4+x^3+x+1)$ 산술 양자 회로 구현 결과

Implementation	Method	Qubits	Clifford gates	T gates	T-depth	Full depth
Quantum-Quantum	Naive	152	784	896	92	147
Classical-Quantum	Naive	24	45	-	-	14
	LUP	16	37	-	-	13

[표 2] 행렬-벡터 곱 인코딩 양자 회로 구현 결과

Berlekamp-Massey decoding	Qubits	Clifford gates	T gates	T-depth	Full depth
mceliece344864	888,492	12,823,392	579,384	60,800	363,696

[표 3] Berlekamp-Massey 디코딩 양자 회로 구현 결과

구현할 수 있다. 선형 연산으로 분류되기 때문에 직관적으로 CNOT 게이트만을 사용하여 구현하거나, LUP 분해를 기반으로 CNOT, Swap 게이트를 사용하여 구현할 수 있다. 하지만 Swap 게이트의 경우, 큐비트의 인덱스를 변경하는 logical Swap이 가능하기 때문에 구현 시 비용을 차지하지 않는다.

곱셈 연산의 경우, 바이너리 필드 산술 중 높은 계산 복잡도를 요구한다. 필드 크기가 n인 경우 일반적인 Schoolbook 곱셈을 수행할 경우, n^2의 AND 연산이 요구된다. 양자 Toffoli 게이트가 고전적인 AND 연산을 수행하는데, 8개의 Clifford + 7개의 T 게이트로 구현되는 Toffoli 게이트는 높은 비용을 차지하는 양자 게이트에 속한다. 때문에 양자 회로 상에서 바이너리 필드 곱셈을 최적화하기 위한 연구들이 다수 존재한다. 본 구현에서는 최근 WISA'22에서 발표된 Toffoli depth가 1로 최적화되는 양자 곱셈 기법 [2]을 적용하여 $\mathbb{F}_{2^{12}}/(x^{12}+x^3+1)$와 $\mathbb{F}_{2^{13}}/(x^{13}+x^4+x^3+x+1)$의 곱셈을 구현한다. WISA'22의 저자들은 카라추바 알고리즘을 재귀적으로 적용, 모든 곱셈의 수행 단위를 1로 줄임으로써 필요 Toffoli 게이트 수를 감소시킨다. 또한 추가 큐빗을 사용함으로써 모든 1단위의 곱셈들을 한 번에 수행함으로써 필드 크기에 상관없이 Toffoli depth가 1로 최적화되며, Full depth 또한 매우 낮은 양자 곱셈이 가능하다. [3]에서는 기본적인 Schoolbook 곱셈을 적용한 양자 곱셈이 제시되었는데, 해당 기법과 [2]의 기법을 구현한 결과를 비교해보았을 때, [2]의 방법이 큐비트를 더 많이 사용하지만, 게이트 복잡도, depth가 훨씬 감소되는 것을 확인할 수 있다.

역치 연산의 경우, Itoh-Tsujii 알고리즘을 기반으로 하여 제곱과 곱셈 연산을 조합하여 구현할 수 있다. 다수의 제곱, 곱셈 연산들이 수행되는 만큼 가장 높은 계산 복잡도가 요구된다. 본 구현에서는 [4]에서 제시된 Itoh-Tsujii 기반의 양자 역치 연산 기법을 적용한다. [4]의 저자들은 역치 연산 내부의 곱셈을 구현하는데 있어 [2]의 기법을 효과적으로 적용하였다. [2]의 기법은 stand-alone 곱셈이 아닌 경우, 많은 보조 큐비트들이 다음 연산에 재사용될 수

있다. [2, 4] 기법을 본 구현에 적용함으로써, 역치 연산 구조에서 첫 번째 곱셈의 보조 큐비트들을 재사용한다. 그 결과, 낮은 depth 그리고 큐비트 수를 감소시킨 효율적인 역치 연산 양자 회로가 구현된다.

2.2 인코딩 양자 회로 구현

Classic McEliece의 인코딩은 바이너리 필드 산술을 사용하여 생성한 공개키 H와 랜덤 벡터 e의 행렬-벡터 곱셈을 수행한다. 공개키 H는 사전에 정해진 값이기 때문에 H의 값에 따라 양자 회로를 구현할 수 있다. 본 논문에서는 세 가지 방법으로 구현하고 회로 비용을 비교한다. 첫 번째는 행렬 H의 값에 따라 CNOT 게이트 (XOR 연산)를 결과 벡터에 수행하는 naive한 구현이다. 두 번째는 행렬 H의 LUP 분해를 기반으로하여 벡터 e에 결과 벡터가 연산되는 in-place 구현이다. 세 번째는 행렬 H와 벡터 e가 모두 양자 상태인 경우에 대한 구현이다. 명시할 점은, 가장 작은 파라미터의 공개키 (768 × 3844) 행렬에 대한 양자 시뮬레이션이 불가능함에 따라 본 구현 및 비용 분석에는 축소된 행렬-벡터 곱셈을 구현한다. [표 2]는 행렬-벡터 곱셈을 양자 회로로서 구현한 결과를 보여준다. 구현 결과에 대해 분석해보면, LUP 분해의 경우 in-place 구현이 가능함에 따라 필요 큐비트 수가 최소화된다. 반면, naive 구현의 경우, 결과 벡터를 위한 큐비트를 할당하기 때문에 큐비트 수가 증가한다. CNOT 게이트 수와 depth 측면으로는 두 방식 모두 유사한 성능을 제공한다. 행렬과 벡터가 모두 양자 상태인 곱셈의 경우, Toffoli 게이트가 사용되어 구현 비용이 가장 높다. 곱셈 대상인 행렬과 벡터 그리고 결과 벡터를 큐비트 상태로 준비하고, 행렬 큐비트와 벡터 큐비트들이 컨트롤 큐비트 역할, 결과 벡터 큐비트가 타겟 큐비트 역할을 한다. 이를 통해 행렬-벡터간의 AND 연산을 수행한 결과를 결과 벡터에 XOR 시킬 수 있다.

2.2.1 인코딩 양자 회로 구현 결과에 대한 분석

행렬-벡터 곱에 대한 양자 구현을 다룬 최신 연구를 보면, LUP 분해가 큐비트 수를 최소화할 수 있는 것은 맞지만, depth가 naive한 구현보다 훨씬 높은 것을 확인할 수 있다. 이는 in-place 구조에서 다수의 CNOT 게이트들이 작동될 경우 순차적인 CNOT 게이트가 강요되기 때문이며, 결과 벡터가 따로 할당되는 경우 (naive) 연산 공간이 넓어지게 되고 다수의 CNOT 게이트들이 병렬로 동작하기 때문이다. 이례적으로, [표 3]의 결과는 작은 행렬에 대한 시뮬레이션이기 때문에 일관적이지 못한 결과라고 평가할 수 있다.

2.3 디코딩 양자 회로 구현

Algorithm 3: 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

$b \leftarrow \mathrm{X}(b[0])$
$C[0] \leftarrow \mathrm{X}(C[0][0])$
$B[1] \leftarrow \mathrm{X}(B[1][0])$
for $N = 0$ to $2t - 1$ **do**
  $d \leftarrow$ new 12-qubit allocation
  for $i = 0$ to min(N, t) **do**
    $d \leftarrow \mathrm{MultiplicationXOR}(C[i], s[N-i], ac)$
  end for
  if $(2L \leq N)$ **then**
    for $i = 0$ to t **do**
      $T[i] \leftarrow$ new 12-qubit allocation
      $T[i] \leftarrow \mathrm{CNOT32}(C[i], T[i])$
    end for
  end if
  $b^{-1} \leftarrow \mathrm{Inversion}(b, ac)$
  if $(2L > N)$ **then**
    for $i = 0$ to t **do**
      $C[i] \leftarrow \mathrm{MultiplicationXOR}(f, B[i], ac)$
    end for
  end if
  if $(2L \leq N)$ **then**
    for $i = 0$ to t **do**
      $C[i] \leftarrow \mathrm{MultiplicationXOR}(f, B[i], ac)$
      $L \leftarrow N + 1 - L$ (classical)
    end for
    for $i = 0$ to t **do**
      $B[i] \leftarrow T[i]$
    end for
    $b = d$ (classical)
  end if
  for $i = 0$ to $t - 1$ **do**
    $B[t-i] \leftarrow B[t-1-i]$
  end for
  $B[0] \leftarrow$ new 12-qubit allocation
end for
return C

[알고리즘 1] Berlekamp-Massey 디코딩 양자 구현

Classic McEliece의 디코딩에서는 인코딩 단계의 결과 벡터로부터 원본 벡터 e를 복구한다. 구체적으로는, 공개키를 생성하는데 사용된 비밀 값들과 Berlekamp-Massey 디코딩 알고리즘을 사용하면 벡터 e의 오류 위치를 (비트 값이 1인 위치) 찾아낼 수 있다. Berlekamp-Massey 디코딩에 대한 자세한 설명은 [5]에서 확인할 수 있다. 본 장에서는 Berlekamp-Massey 디코딩 양자 회로 구현에 대해 설명하며 mceliece348864 파라미터를 대상으로 한다. 제안하는 Berlekamp-Massey 디코딩 양자 회로는 [알고리즘 1]과 같으며, 결과

벡터 s로부터 오류위치 다항식인 C를 찾아내며 핵심이 되는 $\mathbb{F}_{2^{12}}/(x^{12}+x^3+1)$의 곱셈, 역치 연산은 앞서 구현한 양자 회로를 사용한다. [알고리즘 1]에서 알 수 있듯이, 큐비트 수를 줄이기 위해 곱셈과 역치 연산이 반복적으로 사용되는 특징을 활용하여 초기 할당한 대량의 보조 큐비트 (ac)를 연산 종료까지 재사용한다. Berlekamp-Massey 디코딩을 위해 사용되는 값들을 [알고리즘 1, line 1~3]을 통해 준비하고, [알고리즘 1, line 4~35]의 반복 과정을 수행함으로써 오류위치 다항식 C를 계산한다. 제안하는 Berlekamp-Massey 디코딩 양자 회로에 대한 구현 결과는 [표 3]과 같다. Classic McEliece의 실제 파라미터인 mceliece344864를 대상으로 하며, 적지 않은 비용의 양자 곱셈, 역치 연산들이 다수 반복됨에 따라 매우 높은 비용의 양자 자원이 사용된다.

III. 결론 및 토론

양자 컴퓨터를 사용하는 잠재적이고 강력한 암호 분석 능력이 대두됨에 따라, 기존 암호 시스템들의 보안성을 재평가하는 것은 안전한 암호 시스템 구축을 위해 필요하다. 본 논문에서는, NIST Round 4 후보 알고리즘의 코드기반 암호 중 하나인 Classic McEliece의 핵심 연산들을 양자 회로 상에서 최적화하여 구현하였다. 최신 구현 기법들을 적용하여 키 생성, 인코딩, 디코딩의 핵심 연산들을 양자 회로로 구현하였으며 특히, Berlekamp-Massey 디코딩 양자 회로의 경우, 본 논문에서 최초로 제시하는 구현 결과이다. 제시하는 양자 회로들을 사용하여 Classic McEliece의 원활한 양자 암호 분석에 기여할 수 있고자 한다.

향후 연구 방향으로는 구현한 핵심 연산들을 기반으로 Classic McEliece 전체 양자 회로를 완성시키는 것이다. Classic McEliece의 키 크기가 매우 큼에 따라 시뮬레이션이 가능한 범위를 조정해야할 수 있을 것으로 예상되며, 핵심 연산들의 최적화된 조합 또한 중요할 것으로 사료된다.

IV. Acknowledgement

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, 100%).

[참고문헌]

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

[2] 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.

[3] 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.

[4] 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.

[5] E. R. Berlekamp, Binary BCH Codes for Correcting Multiple Errors, Algebraic Coding Theory, 2015.