그루버 알고리즘 적용을 위한 LEA 양자 회로 최적화

장경배*, 김현준*, 박재훈*, 서화정*†
*한성대학교 IT 융합공학부
starj1023@gmail.com, khj93072004@gmail.com, p9595jh@gmail.com,
hwajeong84@gmail.com

Optimization for LEA Quantum Circuit for Applying Grover's Algorithm

Kyung-Bae Jang*, Hyun-Jun Kim*, Jae-Hoon Park* Hwa-Jeong Seo*†
*Dept. of IT Convergence Engineering, Hansung University

요 약

양자 컴퓨터를 활용한 양자 알고리즘은 우리가 현재 사용하고 있는 많은 암호들의 안전성을 깨뜨릴 수 있다. 그루버 알고리즘을 n-bit 보안레벨을 가지는 대칭키 암호에 적용한다면 보안레벨을 $O(2^{n/2})$ 까지 낮출 수 있다. 그루버 알고리즘을 적용하기 위해서는 우선 대상 암호가 양자 회로로 구현되어야 한다. 때문에 대상 블록암호를 양자 회로로 최적화하는 연구들이 최근 활발히 진행되고 있다. 이에 본 논문에서는 국산 경량 블록암호 LEA를 양자 회로에서 최적화 하였다. 기존의 LEA 양자 회로 구현과 비교하여 양자 게이트는 더 많이 사용하였지만, 큐빗을 획기적으로 줄일 수 있었으며 이에 대한 성능 평가를 수행하였다. 마지막으로 제안하는 LEA 구현에 그루버 알고리즘을 적용하기 위한 양자 자원을 평가하였다.

1. 서론

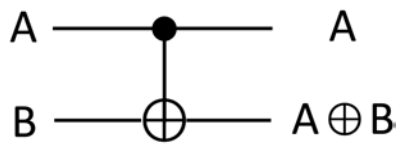
IoT(Internet of Things) 기술이 발전함에 따라 수많은 스마트 디바이스들이 일상생활 속에 자리 잡고 있다[1]. 그리고 IoT 디바이스들은 단순한 센서 데이터뿐만 아니라 개인에게 민감한 정보들까지 포함하여 서로 통신하고 있다. 개인정보 침해의 문제를 막기 위해서는 디바이스들의 통신 과정이 제 3자로부터 보호되어야 한다. 이러한 보안성을 달성하기 위해 암호 알고리즘이 데이터 보호에 적용된다. 암호 알고리즘은 적정 계산 능력과 메모리가 요구된다. 하지만 작은 메모리와 낮은 성능의 IoT 디바이스에는 일반 컴퓨터에 적용되는 암호 알고리즘을 사용하기에는 무리가 있다. 이러한 상황을 해결하기 위해, 국내에서는 저전력 디바이스를 대상으로 한 경량 블록 암호 LEA[2], CHAM[3], HIGHT[4] 가 개발되었다.

암호 알고리즘은 고전 컴퓨터의 공격에 대해서 안전해야 하는 것은 물론이며 양자 컴퓨터와 양자 알고리즘의 공격에 대해서도 내성을 가져야 한다. 양자 알고리즘인 그루버 알고리즘은 n-bit 보안레벨을 가지는 대칭키 암호 알고리즘에 대하여 $O(2^{n/2})$ 보안레벨까지 낮출 수 있는 블록암호에 대한 가장 효과적인 공격 방법이다[5]. 양자 컴퓨터는 아직 개발 초기 단계이기 때문에 그루버 알고리즘을 활용하기 위한 최적의 자원을 찾는 것이 중요하며 최근에 많은 연구가 진행되고 있다. [6]은 블록암호 AES를 양자 회로로 구현하여 그루버 알고리즘을 적용하는데 필요한 양자 자원을 추정하였고, [7,8]은 앞선 연구보다 최적화 된 AES 양자 회로를 제안하였다. [9]은 미국 NSA(National Security Agency)에서 개발한 경량 블록암호 SIMON을 양자 회로로 구현하였으며 [10]에서는 SPECK을 양자 회로로 구현하였다. 마지막으로 [11]은 국산 경량 블록 암호 HIGHT, LEA, CHAM을 양자 회로로 구현 하였다. 본 논문에서는 국산 경량 블록 암호 LEA를 양자 회로로 최적화 구현하여 그루버 알고리즘 적용 자원을 분석하였다. [11]의 LEA 구현보다 양자 게이트를 더 많이 사용하지만, 사용 큐빗의 개수를 획기적으로 줄였다.

2. 관련연구

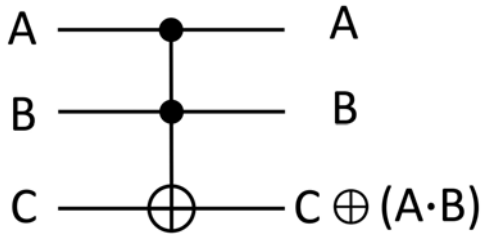
2.1 양자 게이트

양자 게이트 중에는 기존 연산과 동일한 역할을 수행하는 몇몇 게이트들이 있다. X 게이트, CNOT 게이트, Toffoli 게이트가 대표적으로 그러하다. X 게이트는 해당 큐빗의 상태를 반전시킨다. CNOT 게이트는 2개의 큐빗을 입력받아 첫 번째 큐빗이 1인 상태라면 두 번째 큐빗의 상태를 반전시킨다. CNOT 게이트는 기존 XOR 연산을 대체하며 그림 1과 같다.



(그림 1) CNOT 게이트

Toffoli 게이트는 3개의 큐빗을 입력받아 첫 번째와 두 번째 큐빗의 상태가 모두 1이라면, 세 번째 큐빗의 상태를 반전시킨다. 기존 AND 연산을 대체할 수 있으며 그림 2와 같다.



(그림 2) Toffoli 게이트

2.1 Grover on Korean Block Ciphers[11]

2020년, 국산 경량 블록암호 LEA, CHAM, HIGHT를 양자 회로로 구현 및 최적화하는 연구가 진행되었다. 3가지 암호 알고리즘 모두 사용 큐빗을 절약하기 위해 라운드 함수와 키 스케줄을 병행하여 수행하는 기법이 적용되었다. 하지만 LEA에서 가장 많은 자원이 사용되는데, 특히 큐빗의 사용 개수가 HIGHT, CHAM과 비교하여 매우 높다. 해당 논문의 구현 결과는 표1과 같다.

〈표 1〉 구현 결과[11]

	Qubit	Toffoli	CNOT	X
CHAM 64/128	196	2,400	12,285	240
CHAM 128/128	268	4,960	26,885	240
CHAM 128/256	396	5,952	32,277	304
HIGHT	201	6,272	20,523	4
LEA 128	385	10,416	28,080	68
LEA 192	513	15,624	39,816	100
LEA 256	641	17,856	45,504	130

LEA의 키 스케줄 함수에서 상수 덧셈이 수행되는데 이때 필요한 상수들을 모두 큐빗에 할당하였기 때문이다. 본 논문에서는 하나의 상수 큐빗 배열을 재활용하며 키 스케줄을 수행한다. 상수를 변환하는 작업에 양자 게이트가 소모되었지만, 다른 상수들을 위한 큐빗을 할당하지 않음으로써 큐빗의 개수를 줄일 수 있었다.

3. 제안 기법

본 논문에서는 LEA 키 스케줄의 상수 덧셈 부분을 최적화 하였다. LEA의 키 스케줄에서 상수 값은 'L', 'E', 'A'를 ASCII 코드로 표현한 δ 를 사용하며 수식 1과 같다.

$$\begin{aligned}\delta[0] &= 0xc3efe9db, & \delta[1] &= 0x44626b02\\ \delta[2] &= 0x79e27c8a, & \delta[3] &= 0x78df30ec\\ \delta[4] &= 0x715ea49e, & \delta[5] &= 0xc785da0a\\ \delta[6] &= 0xe04ef22a, & \delta[7] &= 0xe5c40957\end{aligned}$$

(수식 1) 상수 값 δ

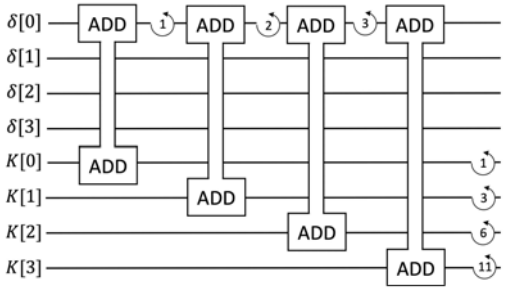
LEA의 키 스케줄에서는 입력 키 값 K와 상수 δ의 모듈러 덧셈 그리고 n-bit 좌측 로테이션 ROL_n 연산을 통해 라운드 키 RK_i를 생성한다. LEA-128의 키 스케줄은 수식 2와 같다.

$$\begin{aligned}K[0] &= ROL_1(K[0] \boxplus ROL_i(\delta[i \bmod 4]))\\ K[1] &= ROL_3(K[1] \boxplus ROL_{i+1}(\delta[i \bmod 4]))\\ K[2] &= ROL_6(K[2] \boxplus ROL_{i+2}(\delta[i \bmod 4]))\\ K[3] &= ROL_{11}(K[3] \boxplus ROL_{i+3}(\delta[i \bmod 4]))\\ RK_i &= (K[0], K[1], K[2], K[1], K[3], K[1])\end{aligned}$$

(수식 2) LEA-128 키 스케줄

기존의 LEA 구현에서는 K에 로테이션 된 δ 의 덧셈을 수행하기 위해, LEA-128의 경우 $\delta[0]$, $\delta[1]$, $\delta[2]$, $\delta[3]$ 값을 위한 큐빗들을 할당하였다. δ 하나당 32 비트로 구성되어 각 32 큐빗이 사용되고 총 128(32×4)개의 큐빗이 사용 되었다. LEA-192은

192개의 큐빗, LEA-256은 256개의 큐빗이 δ 값 선언에 사용되었다. 때문에 CHAM, HIGHT보다 훨씬 많은 큐빗들이 사용되었다. LEA-128의 첫 라운드 키 스케줄에 대한 양자 회로는 그림 3과 같다.



(그림 3) 기존 LEA-128 키 스케줄 양자회로

사용 가능 큐빗의 개수를 늘리는 것은 양자 컴퓨터 개발에 있어 핵심 쟁점이다. 따라서 한정된 큐빗 안에서 계산을 진행해야 하기 때문에 구현 시 큐빗의 개수를 줄이는 것이 매우 중요하다.

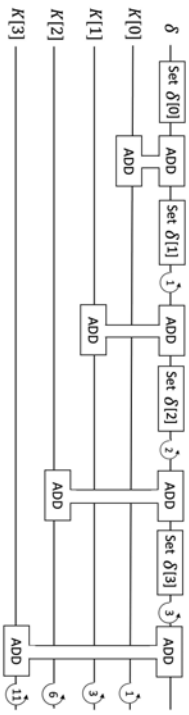
본 논문에서는 기존 최대 8개의 δ[0~8]을 사용하는 경우에도 하나의 δ 값만 사용한다. 사용되는 δ의 값과 순서는 정해져 있다. 따라서 사전에 키 스케줄의 연산에 필요한 δ값을 X 게이트만을 활용하여 만들 수 있다. 동작 예는 다음과 같다. δ[0]가 가장 먼저 사용되기 때문에 선언한 32 큐빗의 δ를 δ[0]의 32비트 값 중 1인 위치에 X 게이트를 통과 시킨다. 예를 들어 δ[0]가 만약 0x00000002 라면 2번째 큐빗에만 X 게이트를 통과시켜 0에서 1로 반전시켜 값을 완성한다. 사용된 후에는 다음에 사용될 δ[1]의 값으로 교체해주어야 한다. 하지만 기존 컴퓨터와 달리 양자 컴퓨터에서는 기존 값을 새로운 값으로 덮어씌우는 것이 불가능하다. 따라서 기존 상태에서 X 게이트를 통과시켜 다음 값을 만들어준다. 만약 δ[1]가 0x00000003 이라면 첫 번째 큐빗에만 X 게이트를 통과시켜 값을 완성해줄 수 있다. 위와 같은 방식으로 키 스케줄에서 사용되는 모든 상수 δ 덧셈을 하나의 32큐빗 배열만을 활용하여 수행한다. 하나의 예를 들어 δ[0]에서 δ[1] 으로 X 게이트로 값을 바꾸는 자세한 과정은 알고리즘 1과 같다. 값을 변경할 때 마다 추가적인 X 게이트를 사용하지만 이는 다른 Toffoli, CNOT 게이트와 비교하여 매우 비용이 낮은 게이트이다. 따라서 조금의 게이트 비용을 더함으로써 기존 연구 결과보다 LEA-128에서는 96개의 큐빗, LEA-256에서는 160개의 큐빗, LEA-256에서는 224개의 큐빗을 절약할 수 있었다.

LEA-128의 첫 라운드 키 스케줄에 대한 양자 회로는 그림 4와 같다. 이에 대한 자세한 비교는 4장에서 다시 살펴보고자 한다.

Algorithm 1 : Change $\delta[0]$ to $\delta[1]$

Input: $\delta_{0\sim31}$ $(0xc3efe9db)$
Output: $\delta_{0\sim31}$ $(0x44626b02)$
1: X gate(δ_0), X gate(δ_3)
2: X gate(δ_4), X gate(δ_6), X gate(δ_7)
3: X gate(δ_9)
4: X gate(δ_{15})
5: X gate(δ_{16}), X gate(δ_{18}), X gate(δ_{19})
6: X gate(δ_{23})
7: X gate(δ_{24}), X gate(δ_{25}), X gate(δ_{26})
8: X gate(δ_{31})
9: **return** $\delta[1] = \{\delta_0, \delta_1, ..., \delta_{31}\}$

(알고리즘 1) $\delta[0]$에서 $\delta[1]$로의 변환



(그림 4) 제안하는 LEA-128 키 스케줄 양자 회로

4. 비교분석

라운드 함수와 키 스케줄을 병행하는 것과, 라운드 함수의 구현은 기존 기법과 동일하다. 하지만 기존 기법에서 키 스케줄에의 상수 값들을 큐빗으로 할당한 반면에, 제안 기법에서는 하나의 32큐빗 배열만을 할당한 뒤, 해당 공간의 값을 채워 넣고 변경해가는 방식으로 키 스케줄의 상수 덧셈을 수행하였다. 따라서 제안하는 기법에서 큐빗의 사용 개수를 획기적으로 줄였다. 제안 기법과 기존 기법의 비교결과는 표 2와 같다. X 게이트의 증가와 큐빗 감소의 트레이드

오프 문제에서 비용이 저렴한 X 게이트로 비용이 비싸고 많은 큐빗의 개수를 줄였기 때문에 제안하는 기법이 더 최적화 잘 되었다고 평가할 수 있다.

〈표 2〉 LEA 양자회로 구현 비교

	Qubit	Toffoli	CNOT	X
LEA 128[11]	385	10,416	28,080	68
LEA 192[11]	513	15,624	39,816	100
LEA 256[11]	641	17,856	45,504	130
Proposed LEA 128	289	10,416	28,080	352
Proposed LEA 192	353	15,624	39,816	398
Proposed LEA 256	417	17,856	45,504	465

5. 결론

현재 블록암호를 양자 회로로 구현하여 그루버 알고리즘을 어떻게 최적화하여 적용할 수 있는지에 대한 연구가 다양하게 진행되고 있다. 블록암호에 그루버 알고리즘을 적용하기 위해 소모되는 자원은 양자 컴퓨터의 공격에 대한 안전성의 지표가 될 수 있다. 이에 본 논문에서는 기존 LEA를 양자 회로로 구현한 기법과 비교하여 양자 게이트의 수는 증가하였지만 큐빗의 사용을 감소시켰다. 따라서 그루버 알고리즘을 적용하는데 필요한 큐빗의 개수를 최적화 할 수 있었다.

참고문헌

[1] Atzori, L.; Iera, A.; Morabito, G. “The Internet of Things: A survey” Computer networks 2010, 54, 2787 - 2805

[2] D. Hong, J.K. Lee, D.C. Kim, D. Kwon, K.H. Ryu, D.G. Lee, “LEA: A 128-bit block cipher for fast encryption on common processors” International Workshop on Information Security Applications. Springer, pp. 3 - 27, 2013.

[3] B. Koo, D. Roh, H. Kim, Y. Jung, D.G. Lee, D. Kwon, “CHAM: A Family of Lightweight Block Ciphers for Resource-Constrained Devices.” International Conference on Information Security and Cryptology (ICISC’17), 2017

[4] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, S. Lee, C. Chang, D. Lee, J. Jeong, K. others. “HIGHT: A new block cipher suitable for low-resource device. International Workshop on Cryptographic Hardware and Embedded Systems.” Springer, pp. 46 - 59, 2006.

[5] Grover, L.K, “A fast quantum mechanical algorithm for database search.” Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 1996, pp. 212 - 219

[6] Grassl. M, Langenberg. B, Roetteler. M, Steinwandt. R, “Applying Grover’s algorithm to AES: quantum resource estimates. Post-Quantum Cryptography.” Springer, pp. 29 - 43, 2016.

[7] Langenberg. B, Pham. H, Steinwandt. R, “Reducing the cost of implementing AES as a quantum circuit.” Technical report, Cryptology ePrint Archive, Report 2019/854, 2019.

[8] Jaques. S, Naehrig. M, Roetteler. M, Virdia. F, “Implementing Grover oracles for quantum key search on AES and LowMC.” Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 280 - 310, 2020.

[9] Anand, R.; Maitra, A.; Mukhopadhyay, S. “Grover on SIMON.” 2020.

[10] K.B. Jang, S.J. Choi, H.D. Kwon, H.J. Seo, “Grover on SPECK : Quantum Resource Estimates.” ePrint Archive, Report 2020/640, 2020.

[11] K.B. Jang, S.J. Choi, H.D. Kwon, H.J. Seo, H.J. Kim, J.H. Park, H.J. Seo, “Grover on Korean Block Ciphers”, Appl. Sci. 10, 6407. 2020.