경량암호 Sparkle에 대한 Grover 공격 비용 분석 및 양자 후 보안 강도 평가

양유진[1], 장경배[1], 송경주[1], 김현지[1], 임세진[1], 서화정[1]
[1]한성대학교 IT융합공학과
yujin.yang34@gmail.com, starj1023@gmail.com, thdrudwn98@gmail.com,
khj1594012@gmail.com, dlatpwls834@gmail.com, hwajeong84@gmail.com

Analysis of Grover Attack Cost and Post-Quantum Security Strength Evaluation for Lightweight Cipher Sparkle

Yu-Jin Yang[1], Kyung-Bae Jang[1], Gyeong-Ju Song[1],
Hyun-ji Kim[1], Se-jin Lim[1], Hwa-jeong Seo[1]
[1]Dept. of IT Convergence, Han-Sung University

요 약

고성능 양자 컴퓨터가 개발될 것으로 기대됨에 따라 잠재적인 양자 컴퓨터의 공격으로부터 안전한 양자 후 보안 시스템을 구축하기 위한 연구들이 진행되고 있다. 대표적인 양자 알고리즘인 Grover 알고리즘이 대칭키 암호에 적용될 경우 보안 강도가 제곱근으로 감소되는 보안 훼손이 발생한다. NIST는 대칭키 암호에 대한 양자 후 보안 요구사항으로, 암호 알고리즘 공격에 요구되는 Grover 알고리즘 비용을 기준을 기준으로 양자 후 보안 강도를 추정하고 있다. 대칭키 암호를 공격하기 위한 Grover 알고리즘의 비용은 대상 암호화 알고리즘의 양자 회로 복잡도에 따라 결정된다. 본 논문에서는 NIST 경량암호 공모전 최종 후보에 오른 Sparkle 알고리즘의 양자 회로를 효율적으로 구현하고 Grover 알고리즘을 적용하기 위한 양자 비용에 대해 분석한다. 마지막으로, NIST 양자 후 보안 요구사항과 분석한 비용을 기반으로 경량암호 Sparkle에 대한 양자 후 보안 강도를 평가한다. 양자 회로 구현 및 비용 분석에는 양자 프로그래밍 툴인 ProjectQ가 사용되었다.

1. 서론

IBM, Google을 선두로 한 국제 IT 대기업들이 고성능 양자 컴퓨터의 개발을 앞당기고 있다. 양자 역학의 특성을 가진 양자 컴퓨터는 특정 문제를 효율적으로 모델링하고 해결할 수 있는데, 현재 암호 시스템들이 기반하고 있는 수학적 난제 또한 빠르게 해결할 수 있을 것으로 예상되고 있다. 이에 암호학계에서는 잠재적인 양자 컴퓨터의 공격으로부터 안전한 양자 후 보안 시스템을 구축하기 위한 움직임을 보이고 있다.

공개키 암호의 경우, 가장 널리 사용되고 있는 RSA와 ECC가 양자 Shor 알고리즘에 의해 다항 시간 내에 해킹될 수 있음에 따라 새로운 양자 내성 암호가 필요한 상황이다. 이에 NIST는 양자내성암호 표준화 공모전을 주최하였으며 현재 몇 개의 후보 알고리즘만이 Finalist로 추려진 상황이다.

대칭키 암호의 경우, 대표적인 양자 알고리즘인 Grover 알고리즘은 비밀키를 찾기 위한 전수 조사 복잡도를 $\sqrt{}$ 만큼 감소시킬 수 있다. 2016년, NIST는 대칭키 암호에 대한 양자 후 보안 요구 사항으로 해당 암호 알고리즘을 공격하는데 사용되는 Grover 알고리즘의 비용을 제시하였다[1]. 이에 본 논문에서는 경량암호 Sparkle에 대한 양자 공격 비용 분석을 수행한다. 이를 위해 Sparkle의 양자 회로를 효율적으로 구현하여 Grover 알고리즘을 적용시킨다. 최종적으로 추정한 비용을 기반으로, NIST가 제시한 보안 요구사항에 따라 Sparkle의 양자 후 보안 강도를 평가한다. NIST 경량암호 공모전의 최종 후보 알고리즘인 Sparkle에 대한 양자 후 보안성 평가를 통해, 안전한 IoT 시스템 구축에 기여하고자 한다.

2. 관련연구

2.1 Grover Search Algorithm

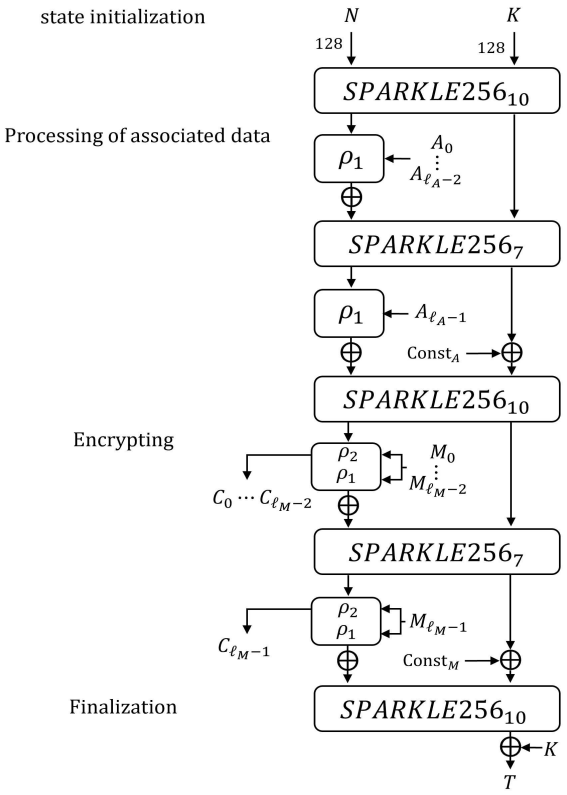
1996년 Lov Grover에 의해 제안된 Grover search algorithm은 양자 역학의 중첩(superposition)과 얽힘(entanglement) 원리를 이용하여 원하는 답을 도출해내는 양자 검색 알고리즘이다[2, 3]. 함수 도메인 크기가 N일 때, Grover 알고리즘을 이용하면 계산

복잡도가 $O(N)$에서 $O(\sqrt{N})$으로 줄어든다.

작동 과정을 예를 들어 설명하면 N=4일 때, Initialization 단계를 거치면 입력 큐비트들이 중첩상태가 된다. 00, 01, 10, 11 중 답이 10일 경우 Oracle function을 이용하여 10에 해당하는 state의 위상을 반전시켜준다. 위상 반전 이후 Amplification 과정을 거치면 모든 state의 평균만큼 반전시키면서 해당 state의 크기가 증폭된다. 이후 이 과정을 $\sqrt{N}$번, 즉 2번 반복하면 오차율은 줄고 원하는 값 10의 크기는 극대화되어 원하는 정답을 도출할 수 있게 된다.

2.2 Sparkle SCHWAEMM-128/128

Sparkle은 NIST LWC Standardization Finalist에 오른 경량 암호 알고리즘으로 Sponge 구조에 기반하여 설계되었다. 64-bit ARX 기반의 S-box인 Alzette를 사용하며 Hash 함수 패밀리에 속하는 ESCH와 AEAD 패밀리에 속하는 SCHWAEMM으로 구성되어 있다[4].



(그림 1) SCHWAEMM-128/128 전체 흐름

SCHWAEMM-128/128은 총 4단계로 상태 초기화(State Initialization), 연관데이터 처리(Processing of associated data), 암호화(Encrypting), 완료(Finalization) 단계를 거친다.

상태 초기화 단계에선 128-bit의 Nonce와 Key를 각각 차례로 합친 내부상태 S를 SPARKLE256에 넣어 Sparkle 암호화 순열 작업을 수행한다.

연관 데이터 처리 단계에서는 128-bit 크기로 패딩 처리된 연관데이터 A를 feedback function ρ 1에 넣어 $(S, D) \mapsto \text{FeistelSwap}(S) \oplus D$ 연산을 수행하고, XOR 연산과 Sparkle 암호화 순열을 거치며 S를 갱신해준다. 이때, 연관데이터가 마지막 블록이 아닐 경우 7 라운드 암호화 순열 과정을 거치고, 마지막 블록일 경우 내부상태의 오른쪽을 나타내는 S_R에 상수 $const_A$를 XOR 연산한 후 10 라운드 암호화 순열을 거친다.

암호화 단계는 입력 메시지 M을 암호화하는 과정으로 $(S, D) \mapsto S \oplus D$ 연산을 수행하는 ρ 2함수와 내부상태의 가장 왼쪽부터 t만큼만 반환하는 $trunc_t$함수를 거쳐 암호문 C가 생성된다. 이도 연관데이터와 마찬가지로 마지막 블록이 아닐 경우 라운드 수를 7로 가지고, 마지막 블록인 경우 라운드 수를 10으로 가지며 S_R에 상수 $const_M$이 XOR 연산된다.

마지막 완료 단계에서 라운드 10번을 도는 SPARKLE256을 수행하여 S를 업데이트 한 후, 키 K와 S_R을 XOR 연산하면 TAG가 생성된다.

그림 1은 해당 알고리즘의 전체 흐름을 그림으로 나타낸 것이다.

3. 제안기법

SPARKLE SCHWAEMM-128/128을 기준으로 양자 회로 구현을 하였고, 메시지 M과 연관데이터 A는 32-bit로 고정하여 실험을 진행하였다.

Algorithm 1 : Quantum circuit implementation of State_Init
Input: carry, K[i], S_R[i], N[i], S_L[i+128] (0≤i≤127)
Output: S = S_L || S_R

```
1: for k = 0 to 127
2:     CNOT (K[k], S_R[k])
3:     CNOT (N[k], S_L[k])
4: SPARKLE256(S, carry, 10)
5: return S = S_L || S_R
```

(알고리즘 1) State Initialization 양자 회로 구현

알고리즘 1의 2, 3번째 라인에서 CNOT게이트를 활용하여 내부상태 S에 각각 오른쪽부터 N과 K를 넣어 내부상태를 초기화하였다.

알고리즘2의 2, 3번째 라인에서는 S_R과 사전에 정의된 상수 $const_A$를 XOR 연산을 진행한다. X게이트를 활용하여 $const$를 $0 \oplus (1 \ll 2)$ 즉, 2로 만들어주고 CNOT게이트를 활용하여 S_R과 $const$를 XOR 연산하였다.

이후, ρ 1 함수를 거친 후, SPARKLE256 함수에

상태 S를 넣어 SPARKLE 순열 라운드를 10번 수행하고 나면 Processing_AD 과정이 마무리된다. 이를 위하여 우선적으로 $\mathsf{FeistelSwap}(S) = S_2 \| (S_2 \oplus S_1)$ 연산을 먼저 수행해야 하는데, 4~9번째 라인이 해당 과정을 구현한 부분으로, SWAP게이트를 활용하여 S_{L1}과 S_{L2}를 스와핑하고 CNOT게이트를 통해 S_{L2}과 S_{L1} XOR 연산을 수행한다. 이에 따른 결과로 S_L과 S_R을 연결한 상태 S가 반환된다.

Algorithm 2 : Quantum circuit implementation of Processing_AD
Input: carry, S_R, S_x[i], S_y[i+32], AD, const ($160 \le i \le 191$)
Output: S = S_L || S_R

```
1: X const[2]
2: for k = 3 to 0
3:     CNOT (const[k], S_R[24+k])
4: for k = 0 to 31
5:     SWAP (S_x[k], S_x[k+64])
6:     SWAP (S_y[k], S_y[k+64])
7: for k = 0 to 31
8:     CNOT (S_x[k+64], S_x[k])
9:     CNOT (S_y[k+64], S_y[k])
10: for k = 0 to 31
11:    CNOT (AD[k], S_x[k+64])
12:    CNOT (AD[k+32], S_y[k+64])
13: SPARKLE256(S, carry, 10)
14: return S = S_L || S_R
```

(알고리즘 2) Processing of Associated data 양자 회로 구현

알고리즘 3은 메시를 암호화하는 Encrypting 과정이다. 1~2번째 라인에서 초기화된 32-qubit 암호문 C에 CNOT 게이트를 활용하여 메시지 M을 넣어주고 3~4번째 라인에서 ρ 2함수와 $trunc_t$함수를 진행한다. 여기서 어차피 $trunc_t$를 거치면 t만큼만 쓰일 것이기 때문에 ρ 2함수를 모두 수행하지 않고 메시지의 길이만큼만 수행하였다. ρ 2함수와 $trunc_t$함수를 동시에 실행함으로써 메시지 블록이 127-bit보다 작을 경우 덜 수행할 수 있도록 하였다.

Algorithm 3 : Quantum circuit implementation of Encrypting
Input: S_L[i], M, C, const, Mlen=32 ($224 \le i \le 255$)
Output: C, S_R

```
1: for k = 0 to Mlen
2:     CNOT (M[k], C[k])
3: for k = 255 to 254-Mlen
4:     CNOT (S_L[k], C[k])
5: X const[1]
6: for k = 3 to 0
7:     CNOT (const[k], S_R[24+k])
8: return C, S_R
```

(알고리즘 3) Encrypting 양자 회로 구현

5~7번째 라인은 알고리즘2에서 수행했던 것처럼 CNOT게이트를 사용하여 상수 $const_M$과 S_R를 XOR 연산하는 과정으로, 따로 $const_M$을 할당하지 않고 연관데이터 처리 과정에서 사용했던 $const$를 재활용함으로써 자원을 절약하였다.

Algorithm 4 : Quantum circuit implementation of Finalization
Input: carry, K[i], S_L, S_R, M, C, Mlen=32 ($0 \le i \le 127$)
Output: result

```
1: ρ 1(S_L, M)
2: SPARKLE256(S, carry, 10)
3: for i = 0 to 127
4:     CNOT (K[i], S_R[i])
5:     CNOT (S_R[i], result[i])
6: for k = 0 to Mlen
7:     CNOT (C[k], result[128+k])
10: return result
```

(알고리즘 4) Finalization 양자 회로 구현

알고리즘 4의 1, 2번째 라인에서 파라미터에 AD 대신 M을 넣어 알고리즘 3에서 제시된 ρ 1 함수와 SPARKLE256을 거친다. 3, 4번째 라인은 TAG를 만드는 과정으로 CNOT게이트를 활용하여 S_R과 K를 XOR 연산한 값을 S_R에 저장한다. 5번째 라인은 CNOT게이트를 활용하여 TAG를 최종반환값인 result에 복사하는 부분이다. 6, 7번째 라인 또한 CNOT게이트를 이용하여 암호문 C를 result에 복사하여 넣는 과정이다.

이후 Finalization을 거쳐 나온 최종 반환값 result를 관측하면 TAG||C의 형태로 나온 암호화 데이터값과 연산에 사용된 자원을 확인할 수 있다.

4. 평가

[표 1]은 SCHWAEMM-128/128에 대한 양자 회로 구현 비용을 나타낸다.

<표 1> Quantum resources for SCHWAEMM-128/128

Qubits	X gates	CNOT gates	Toffoli gates	Depth
808	35,949	103,240	29,280	8,601

Grover 알고리즘은 Oracle과 Diffusion operator의 반복으로 구성되지만 Diffusion operator의 경우 Oracle와 비교하여 오버헤드가 매우 적기 때문에 비용 추정 시 일반적으로 무시된다. 따라서 Grover 알고리즘에 대한 공격 비용은 Oracle에 자리하는 암호화 양자 회로를 얼마나 효율적으로 구현하는지에 따

라 결정된다. Oracle은 암호화를 위한 Sparkle 양자 회로 1번, reverse 연산을 위한 Sparkle 양자 회로 1번이 순차적으로 동작한다. 따라서 Oracle 한 번에 대한 비용은 큐비트를 제외한 [표 1] × 2로 계산된다. 128-bit key를 사용하는 Sparkle에 대한 Grover 공격 시, Oracle이 약 $\sqrt{2^{128}}$번 반복되기 때문에 최종 비용은 [표 1] × 2 × 2^{64}로 추정하며 [표 2]와 같다.

<표 2> Cost of Grover key search for SCHWAEMM-128/128

Qubits	Total gates	Total depth	Cost	NIST Secuirty
809	1.74×2^{83}	1.65×2^{77}	1.43×2^{161}	Not achieved (Level 1: 2^{170})

NIST의 양자 후 보안 강도 기준과 Sparkle에 대한 Grover 공격 비용을 비교한다면, 가장 낮은 보안 강도인 Level 1을 달성할 수 없다. 하지만 짚고 넘어가야 할 점은, NIST가 인용하고 있는 공격 비용은 2016년 Grassl et al.의 AES 양자 공격 연구[5]라는 것이다. 해당 연구는 블록암호에 대한 최초의 양자 공격 연구였으며 AES 양자 회로가 매우 비효율적으로 구현되었기 때문에 공격 비용이 매우 높다. NIST 또한 추후 공격 비용이 크게 감소된 양자 공격이 등장하는 경우, 추정한 비용(2^{170})은 보수적으로 간주되어야 한다고 언급하였다. 이후 비용을 감소시키는 다양한 AES 양자 회로 구현 연구들이 등장하였기 때문에[6], [표 2]에서 Sparkle은 Level-1을 달성하지 못하지만 이러한 결과는 보수적으로 평가되어야 한다.

5. 결론

가까운 미래에 등장할 고성능의 양자 컴퓨터로부터 안전한 암호 시스템을 구축하기 위한 확실한 방법은 다양한 암호 알고리즘에 대한 양자 후 안전성을 사전에 평가하는 것이다. 본 논문에서는 경량 암호 Sparkle의 AEAD Family에 해당하는 SCHWAEMM-128/128를 양자 회로로 최적화 구현하였고 Grover 알고리즘 적용에 필요로 하는 비용을 추정하여 NIST가 제시한 보안 요구사항을 기준으로 양자 후 보안성을 평가하였다. 그 결과, 구현한 양자 회로는 보안 강도 Level-1을 달성하지 못했다. 그러나 NIST의 추정 비용을 보수적인 지표로 고려한다면 구현한 Sparkle 양자 회로도 해당 기준을 만족했다고 볼 수 있을 것이다.

향후 연구 방향으로는, NIST 경량암호 공모전의 다양한 최종 후보 알고리즘들에 대한 Grover 공격 비용을 분석하고 양자 후 보안 강도를 평가하는 것이다. 다가오는 양자 컴퓨터 시대에 대비하여 안전한 IoT 시스템을 구축하는 것은, 암호 학계에 있어 중요한 향후 계획이 될 것으로 사료된다.

6. Acknowledgement

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (<Q|Crypton>, No.2019-0-00033, 미래컴퓨팅 환경에 대비한 계산 복잡도 기반 암호 안전성 검증 기술개발, 100%).

참고문헌

[1] NIST, “Submission requirements and evaluation criteria for the post-quantum cryptography standardization process,” [internet], https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

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

[3] B.I. Kim, K.S. Min, J. Heo. "Hamiltonian path problem approach using Grover search algorithm", The Journal of Communications and Networks, Vol. 2020, No. 8, pp. 52-53, 2020.

[4] Beierle, Christof, et al. "Schwaemm and Esch: lightweight authenticated encryption and hashing using the Sparkle permutation family." NIST round 2 (2019).

[5] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, “Applying Grover’s algorithm to AES: quantum resource estimates,” Post-Quantum Cryptography, PQCrypto’16, LNCS, 9606, pp. 29 - 43, 2016.

[6] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, “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.