SPECK 양자 회로 최적화를 통한 양자 후 보안 강도 평가

장경배*, 엄시우*, 송경주*, 양유진*, 서화정*
*한성대학교 IT융합공학부
starj1023@gmail.com, shuraatum@gmail.com, thdrudwn98@gmail.com,
yujin.yang34@gmail.com, hwajeong84@gmail.com

Post-Quantum Security Evaluation Through SPECK Quantum Circuit Optimization

Kyung-Bae Jang*, Si-Woo Eum*, Gyeong-Ju Song*, Yu-jin Yang*,
Hwa-Jeong Seo*
*Dept. of IT Convergence Engineering, Han-Sung University

요 약

양자 알고리즘이 수행 가능한 양자 컴퓨터는 기존 암호 시스템의 보안성을 낮추거나 깨뜨릴 수 있다. 이에 양자 컴퓨터의 공격 관점에서 기존 암호 시스템의 보안성을 재평가하는 연구들이 활발히 수행되고 있다. NIST는 대칭키 암호 시스템에 대한 양자 후 보안 강도에 평가에 Grover 알고리즘의 적용 비용을 채택하고 있다. Grover 알고리즘이 대칭키 암호 시스템의 보안성을 절반으로 줄일 수 있는 시점에서 중요한 건 공격 비용이다. 본 논문에서는 경량블록암호 SPECK 양자 회로 최적화 구현을 제시한다. ARX 구조의 SPECK에 대해 최적의 양자 덧셈기를 채택하고 병렬 덧셈을 수행한다. 그 결과, 최신 구현물과 비교하여 depth 측면에서 56%의 성능향상을 제공한다. 최종적으로, 제시하는 SPECK 양자 회로를 기반으로 Grover 알고리즘 적용 비용을 추정하고 양자 후 보안 강도를 평가한다.

1. 서론

IBM, Google Microsoft, Amazon 등 국제 대기업에서 양자 컴퓨팅 및 양자 프로세서 개발에 전폭적인 투자를 하고 있는 시기이다. 양자 알고리즘을 활용한 양자 컴퓨터는 몇몇 문제를 해결하는 데 있어 기존 컴퓨터보다 뛰어난 계산 능력을 보여주며 현재 암호 시스템의 안전성에도 영향을 끼치고 있다. 대표적인 양자 알고리즘인 Grover search 알고리즘은 정렬되지 않은 N개의 데이터 중 특정 데이터를 $\sqrt{N}$번 만에 높은 확률로 찾아낼 수 있다[1]. Grover search 알고리즘은 대칭키 암호 시스템의 키를 찾아내는 Brute force 공격에 적용될 수 있다. 이를 위해서는 대상 암호의 암호화 연산들이 양자 컴퓨터에서 동작하도록 양자 게이트들로 구현되어야 한다. 이러한 연구 동기에 따라, 대칭키 암호 AES의 양자 회로 구현을 시작으로[2], 다양한 대칭키 암호를 양자 회로로 최적화 구현하는 연구들이 활발히 진행되고 있다. 양자 컴퓨터상에서의 최적화 구현 요소는 큐비트, 양자 게이트, 회로 depth 총 3가지가 있다. 대규모 큐비트의 양자 컴퓨터가 개발되지 않은 현재, 큐비트의 수는 실제 양자 컴퓨터에서 동작되는 시기와 연결되기 때문에 중요한 요소이다. 사용 된 양자 게이트 그리고 depth는 회로의 복잡도를 나타내기 때문에 실행 속도와 연관된다. 현재 개발 수준의 양자 프로세서는 연산에서의 오류 및 큐비트 간 불안전성이 존재한다. 높은 depth와 양자 게이트들로 구성된 양자 회로는 동작 중 생기는 오류를 제어하기 어렵다.

2020년, 경량 블록암호 SPECK이 양자 회로로서 최초로 구현되었으며[2], 2021년 [3]에서 회로 depth가 개선된 SPECK 양자 회로를 제시하였다. 본 논문에서는 SPECK을 양자 회로로 최적화 구현을 제시하여 본 구현은 최신 구현 결과[3]와 비교하여 depth 측면에서 56%의 성능 향상을 제공한다. ARX(Addition Rotation XOR) 구조인 블록암호 SPECK의 덧셈 연산에 최적화된 양자 덧셈기를 채택하여 1차적으로 성능을 향상 시킨다. 또한 제안하는 구현은 라운드 함수와 키 스케줄의 on-the-fly 방식을 채택하고 양자 덧셈을 병렬로 수행함으로써 회로 depth를 크게 줄인다. 마지막으로 제시하는 SPECK 양자 회로를 기반으로 Grover search 알고리즘 적용 비용을 추정한 뒤, NIST에서 제시하는 기준에 맞추어 양자 후 보안 강도를 평가한다.

2. 관련 연구

2.1 SPECK

2013년 미국 NSA(National Security Agency)는 저전력 디바이스를 위한 경량 블록암호군 SPECK과 SIMON을 개발하였다. SPECK은 소프트웨어에 친화적, SIMON은 하드웨어 친화적인 구조로 설계되었다. 본 장에서는 SPECK의 암호화 구조에 대해 설명한다. ARX 구조의 SPECK은 덧셈, Rotation, XOR 연산만으로 구성되어 있다. 암호화 연산의 간단함으로 SPECK은 대부분의 플랫폼에서 준수한 성능을 제공한다. SPECK은 다양한 파라미터를 제공하며 표 1과 같다.

<표 1> SPECK 파라미터

Block size	Key size	Word size(n)	Keywords	Rounds(r)
32	64	16	4	22
48	72, 96	24	3, 4	22, 23
64	96, 128	32	3, 4	26, 27
96	96, 144	48	2, 3	28, 29
128	128, 192, 256	64	2, 3, 4	32, 33, 34

SPECK의 라운드 함수는 $2n$-bit 블록 (x, y)에 대해 동작하며 n-bit 라운드 키가 사용된다. SPECK의 라운드 함수 연산은 다음과 같다.

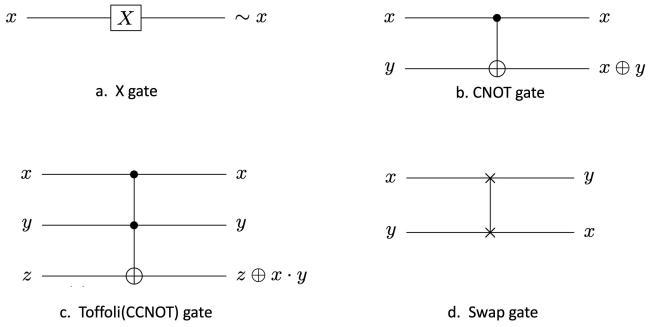
$$R_k(x, y) = ((S^{-\alpha}x + y) \oplus k, S^{+\beta}y \oplus (S^{-\alpha}x + y) \oplus k)$$

S^+는 좌측 Rotation, S^-는 우측 Rotation을 의미한다. α와 β는 32-bit 블록 크기의 경우 각각 7과 3이며 이외의 블록 크기에 대해서는 각각 8과 3이다. k는 n-bit 라운드 키이며 키 스케줄을 수행하여 생성한다. m은 키워드 수를 의미하며 초기 키워드는 $K = (k_0, l_0, \cdots, l_{m-2})$이다. 라운드 키 k_i를 생성하기 위해 다음 키 스케줄을 수행한다.

$$l_{i+m-1} = (k_i + S^{-\alpha}l_i) \oplus i,$$
$$k_{i+1} = S^{+\beta}k_i \oplus l_{i+m-1}$$

2.2 양자 컴퓨팅

양자 회로에서 관측을 제외한 모든 양자 게이트들은 가역적인 특징을 가지고 있다. 즉, 출력 값만을 사용하여 입력 값이 복구될 수 있어야 함을 의미한다. 암호학 연산들을 양자 회로 상에 구현하기 위해 필요한 몇 가지 대표적인 양자 게이트들이 그림 1에 나타나있다. X 게이트(a)는 입력 큐비트를 반전시키며 NOT 연산을 대체할 수 있다. CNOT 게이트(b)는 control 큐비트(x)가 1인 경우에 대상 큐비트(y)를 반전시키며 XOR 연산을 대체할 수 있다. Toffoli gate(c)는 2개의 control 큐비트들이(x, y) 모두 1인 경우 대상 큐비트(z)를 반전시키며 AND 연산을 대체할 수 있다. Swap 게이트는 입력 큐비트의 상태를 서로 변경시킨다. Swap 게이트를 사용하여 Rotation 연산을 수행할 수 있다.



(그림 1) 양자 게이트

2.3 Grover 알고리즘을 활용한 키 복구

Grover search 알고리즘은 알려진 평문-암호문 쌍에 대한 솔루션 키를 찾는 oracle 그리고 찾은 솔루션 키의 amplitude를 증폭 시키는 diffusion operator로 구성된다. Hadamard 게이트를 통과한 n-qubit 키는 중첩 상태가 되어 모든 키 값들이 동일한 amplitude로 존재하게 된다. oracle에는 구현한 암호화 회로가 자리하고 중첩 상태의 키로 입력 평문을 암호화한다. 모든 키 값에 대한 모든 암호문이 한 번에 생성되고 사전에 알고 있는 암호문과 일치하는 경우 Controlled-Z 게이트를 사용하여 솔루션 키의 amplitude의 부호를 음수로 변경한다. 이것이 oracle에서 솔루션 키를 반환하는 방식이며 아직 동일한 amplitude를 가지고 있는 상태이다. 다음 diffusion operator에서는 oracle에서 반환한 솔루션 키의 amplitude를 증폭시킨다. diffusion operator는 형식화된 구조가 존재하여 특별한 구현 기법이 필요하지 않다. 또한 최종 비용은 oracle에 의해 결정되기 때문에 Grover 알고리즘의 비용 추정 시 대부분 oracle에 대한 비용만을 고려한다. oracle과 diffusion operator를 반복 수행하여 솔루션 키의 amplitude를 충분히 높인 뒤, 높은 확률로 솔루션 키를 관측한다. n-qubit 키의 경우 약 $2^{n/2}$번 반복이 필요하고 기존 $O(2^n)$보다 더 빠르게 키를 복구할 수 있다. 후에 Grover 알고리즘에 대한 상세한 분석 결과, 최적의 반복 횟수는 $\frac{\pi}{4}2^{n/2}$번인 것으로 나타났다.

3. 제안 기법

본 장에서는 제안하는 SPECK 양자 회로에 대한 구현 기법에 대하여 설명한다. 덧셈 연산에는 ripple-carry 방식을 사용한 양자 덧셈기가 사용된다. SPECK은 모듈러 덧셈이 사용되고 이 경우 일차적으로 향상된 ripple-carry 양자 덧셈기 구현이 가능하다. 또한 덧셈 단위가 4-bit 이상인 경우, 이차적으로 최적화 된 양자 덧셈기를 사용할 수 있다. SPECK 파라미터들 중 가장 작은 단위의 덧셈은 16-bit 모듈러 덧셈이기 때문에 SPECK의 모든 버전에 적용 가능하다. 마지막으로 on-the-fly 방식을 활용하여 라운드 함수의 덧셈과 키 스케줄의 덧셈을 병렬로 수행한다. 병렬 덧셈을 위해서는 양자 덧셈기에 사용되는 carry 큐비트를 하나 더 할당해야 하지만 회로 depth를 획기적으로 감소시킬 수 있다.

3.1 병렬 덧셈을 통한 SPECK 양자 회로 최적화

우리는 초키 키워드 k_0는 첫 번째 라운드 함수를 위한 라운드 키로 사용하고 k_0를 업데이트해 가며 각 라운드 함수의 라운드 키 k_i로 사용한다. 이를 통해 라운드 키 생성을 위한 추가 큐비트들을 할당하지 않아도 된다. 각 라운드에서는 라운드 함수와 키 스케줄을 함께 수행한다. 이로 인해 라운드 함수의 $S^{-\alpha}x+y$ 와 키 스케줄의 $k_i+S^{-\alpha}l_i$ 덧셈을 병렬로 수행할 수 있다. [2]에서도 동일한 on-the-fly 방식을 사용하였지만 라운드 함수 후 키 스케줄이 수행되어 덧셈이 순차적으로 실행된다. 제안하는 SPECK 양자 회로는 병렬 덧셈을 위해 두 가지 기법이 사용된다.

첫 번째, 각 라운드는 라운드 함수(1/2) → 키 스케줄(1/2) → 라운드 함수(2/2) → 키 스케줄 (2/2) 구조로 설계된다. 라운드 키로 사용되는 k_i는 라운드 함수에서 사용되기 전까지는 다음 라운드 키로 업데이트되면 안 된다. 라운드 함수 (1/2)는 $S^{-\alpha}x+y$ 이며 키 스케줄 (1/2)는 $k_i+S^{-\alpha}l_i$ 이다. k_i는 유지되어야 하기 때문에 키 스케줄의 덧셈 결과는 우선 l_i에 저장한다 ($l_i=k_i+S^{-\alpha}l_i$). 라운드 함수(2/2)와 키 스케줄(2/2)는 남은 모든 연산들을 포함한다.

두 번째, 이제 병렬 덧셈이 가능하기 때문에 우리는 이를 위한 추가 carry 큐비트가 필요하다. 이전 구현[2]에서는 하나의 carry 큐비트만을 선언하여 모든 덧셈에서 재활용한다. 하지만 우리는 한 개의 추가 큐비트를 사용함으로써 2개의 덧셈을 동시에 수행하고 다음 2개의 덧셈에서 재사용한다.

최종적으로 제안하는 양자 회로는 회로 depth 측면에서 56%의 성능 향상을 제공한다. 알고리즘 1은 제시하는 SPECK-32/64에 대한 양자 회로 구현을 설명한다. 해당 기법은 연산 단위, 파라미터만 변경하여 모든 SPECK 버전들에 적용할 수 있다. 추가적으로 Rotation 연산은 Swap 게이트를 사용하여 구현할 수 있지만 우리는 큐비트의 인덱스를 직접 변경하는 논리적 Swap을 사용하여 양자 게이트가 전혀 사용되지 않는다. Addition(a,b,c)의 덧셈 결과는 b 큐비트에 저장된다. CNOT16은 16-qubit 단위의 CNOT 게이트 연산을 의미하며 CNOT(a,b)의 결과는 $a=a,\ \ b=a\oplus b$ 이다.

Algorithm 1 : Quantum circuit for SPECK-32/64

Input: 32-qubit block (x,y), 64-qubit keywords (k_0,l_0,l_1,l_2), Carry qubits c_0,c_1

Output: 32-qubit ciphertext (x,y)

for $i=0$ to $r-2$
  Round function(1/2) :
    $x \leftarrow S^{-7}$
    $x \leftarrow$ Addition(y,x,c_0) //Parallel addition
  Key schedule(1/2) :
    $l_{i\%3} \leftarrow S^{-7}l_{i\%3}$
    $l_{i\%3} \leftarrow$ Addition$(k_0,l_{i\%3},c_1)$ //Parallel addition
  Round function(2/2) :
    $x \leftarrow$ CNOT16(k_0,x)
    $y \leftarrow S^{+2}$
    $y \leftarrow$ CNOT16(x,y)
  Key Schedule(2/2) :
    for $j=0$ to 5 //Constant XOR
      if $(i \gg 1)$ & 1
        $l_{i\%3}[j] \leftarrow$ X$(l_{i\%3}[j])$
    $k_0 \leftarrow S^{+2}k_0$
    $k_0 \leftarrow$ CNOT16$(l_{i\%3},k_0)$
//Last round
Round function(1/2) :
  $x \leftarrow S^{-7}$
  $x \leftarrow$ Addition(y,x,c_0)
Round function(1/2) :
  $x \leftarrow$ CNOT16(k_0,x)
  $y \leftarrow S^{+2}$
  $y \leftarrow$ CNOT16(x,y)
return (x,y)

4. 성능 평가

본 장에서는 제안하는 SPECK 양자 회로의 성능에 대해 평가한다. [2]에서 SPECK의 양자 회로 구현이 처음으로 제시되었으며, [3]은 [2]보다 개선된 SPECK 양자 회로를 제시하였다. 표 2는 [3]에서 제시한 SPECK 양자 회로 구현에 필요한 자원들을 나타낸다. 표 3은 본 논문에서 제시하는 SPECK 양자 회로 구현에 필요한 자원들을 나타낸다. 모든 SPECK 버전들을 구현하였지만 3개의 버전들에 대해서만 대표적으로 비교하도록 한다.

<표 2> [3]의 SPECK 양자 회로 구현에 필요한 비용

SPECK	Qubits	Toffoli	CNOT	X	Depth
32/64	96	1,290	4,222	42	1,694
64/128	192	3,286	10,722	57	4,239
128/256	384	8,442	27,502	81	10,778

<표 3> 제시하는 SPECK 양자 회로 구현에 필요한 비용

SPECK	Qubits	Toffoli	CNOT	X	Depth
32/64	98	1,247	4,179	1,160	814
64/128	194	3,233	10,669	3,131	1,863
128/256	386	8,375	27,435	8,255	4,522

본 양자 회로 구현에 사용된 ripple-carry 양자 덧셈기는 덧셈 시 하나의 carry 큐비트가 필요하고 병렬화를 위해 우리는 2개의 carry 큐비트를 사용한다. 표 2 보다 2개 더 많은 큐비트를 사용하고 X 게이트를 더 많이 사용하지만 양자 게이트 중 높은 비용의 Toffoli 게이트 수는 줄어든다. 적은 depth를 차지하며 수행되는 덧셈 또한 병렬로 수행되기 때문에 최종적으로 depth 측면에서 56%의 성능 향상을 제공한다.

5. Grover 알고리즘 적용 비용 분석

본 장에서는 제시하는 SPECK 양자 회로를 기반으로 Grover 알고리즘 적용 비용을 분석한다. Grover 알고리즘 비용 추정에는 2.3장에서 언급한대로 oracle에 필요한 비용을 추정한다. oracle의 동작 순서로는 제일 먼저 암호화 회로가 작동된다. 그 다음 생성된 암호문이 일치하는지 비교하는데 이 때 다중 CNOT 게이트가 1번 사용되지만 비중이 적고 분석의 간결함을 위해 이에 대한 비용은 제외하도록 한다. 마지막으로 다음 반복을 위해 앞서 수행한 암호화 회로를 reverse로 수행하여 다시 입력 평문 상태로 되돌린다. 즉 oracle 1번에 SPECK 양자 회로가 2번 순차적으로 작동하기 때문에 큐비트를 제외한 (표 2 × 2) 가 된다. 유일한 키를 찾기 위해서는 r = (키 크기)/(블록 크기)의 알려진 평문-암호문 r개의 쌍이 필요하다. 이는 병렬화가 가능하기 때문에 최종적으로 oracle에 필요한 비용은 (표 2 × 2(큐비트 제외)) × r(depth 제외))로 추정할 수 있다. 구체적인 비용 추정을 위해 Toffoli 게이트를 분해해야 하는데 우리는 1개의 Toffoli 게이트를 8 Clifford 게이트 + 7 T 게이트로 분해하여 계산한다. CNOT, X 게이트는 Clifford 게이트로 계산한다. 마지막으로 n-bit 키의 경우 $\frac{\pi}{4}2^{n/2}$번의 oracle 반복 수행이 필요하다. 최종적으로 Grover 알고리즘 적용에는 $\frac{\pi}{4}2^{n/2}$ × (표 2 × 2(큐비트 제외)) × r(depth 제외)) 비용이 필요하다. NIST에서는 이를 기반으로 대칭키 암호 시스템의 양자 후 보안 강도를 평가하고 있으며 [4]에 자세히 나타나 있다. 표 4는 SPECK에 대한 Grover 적용 비용과 이를 기반으로 한 양자 후 보안 강도를 평가한 결과이다.

<표 4> Grover 적용 비용 및 양자 후 보안 강도 평가

SPECK	Qubits	Total gates	Total depth	Cost	NIST security
32/64	132	$1{,}797\times2^{47}$	$1{,}249\times2^{42}$	$1{,}122\times2^{90}$	Not achieved
64/128	389	$1{,}505\times2^{81}$	$1{,}429\times2^{75}$	$1{,}075\times2^{157}$	
128/256	773	$1{,}945\times2^{146}$	$1{,}734\times2^{140}$	$1{,}687\times2^{287}$	Level 3 (2^{233})

6. 결론

본 논문에서는 SPECK에 대한 양자 회로 최적화 구현을 기반으로 Grover 알고리즘 적용 비용을 추정하였으며 다가오는 양자 컴퓨터 시대에 대한 양자 암호 분석에 기여할 수 있다.

7. Acknowledgment

이 논문은 부분적으로 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천 기술 연구, 50%) 그리고 부분적으로 2021년도 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, 50%).

참고문헌

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

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

[3] Anand. R, Maitra. A, Mukhopadhya. S, “Evaluation of quantum cryptanalysis on speck,” Progress in Cryptology INDOCRYPT 2020.

[4] NIST. “Submission requirements and evaluation criteria for the post-quantum cryptography standardization process”, 2016