석사학위논문

블록암호 ARIA에 대한 depth 최적화된 양자 회로 구현

2024년

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

양 유 진

석 사 학 위 논 문
지도교수 서화정

블록암호 ARIA에 대한 depth 최적화된 양자 회로 구현

Depth-Optimized Quantum Circuit Implementation for Block Cipher ARIA

2023년 12월 일

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

양 유 진

석 사 학 위 논 문
지도교수 서화정

블록암호 ARIA에 대한 depth 최적화된 양자 회로 구현

Depth-Optimized Quantum Circuit Implementation for Block Cipher ARIA

위 논문을 공학 석사학위 논문으로 제출함

2023년 12월 일

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

양 유 진

양유진의 공학 석사학위 논문을 인준함

2023년 12월 일

심사위원장 박 명 서 (인)

심 사 위 원 서 화 정 (인)

심 사 위 원 이 웅 희 (인)

국 문 초 록

블록암호 ARIA에 대한
depth 최적화된 양자 회로 구현

한 성 대 학 교 대 학 원
I T 융 합 공 학 과
I T 융 합 공 학 전 공
양 유 진

대규모 양자 컴퓨터의 발전은 현재 암호 시스템의 보안에 위협이 되고 있다. 특히 대칭키 암호는 Grover 검색 알고리즘을 사용하는 일반적인 공격에 의해 큰 영향을 받는다. 최근 몇 년간 대칭키 암호에 대한 Grover 키 검색 복잡성을 추정하고 양자 보안 강도를 평가하기 위한 연구가 발표되어왔다. 본 논문에서는 국내암호모듈검증제도(KCMVP)의 검증 대상으로 포함된 대칭키 암호 ARIA에 대해 depth 최적화된 양자 회로 구현을 제안한다. ARIA에 대한 양자 회로 구현은 SPACE'20에 발표된 Chauhan et al.의 논문에 제시된 구현과 비교하면 depth를 88.8% 이상, Toffoli-depth를 98.7% 이상 향상시켰다. 마지막으로, 제안한 회로에 대한 Grover 키 검색 비용을 제시하고 NIST에서 제공한 관련 평가 기준에 따라 ARIA의 양자 보안 강도를 평가한다.

【주요어】ARIA 블록 암호, Depth 최적화된 양자 회로 구현, Grover 검색

알고리즘, 양자 보안 강도 평가

목 차

표 목 차

그 림 목 차

수 식 목 차

알고리즘 목 차

Ⅰ. 서론

1.1 양자 컴퓨터와 암호

양자 중첩(superposition) 및 얽힘(entanglement)과 같은 양자 역학의 원리를 기반으로 구축된 양자 컴퓨터는 기존 컴퓨터에서 가장 작은 정보 단위를 의미하는 비트(bit)와 유사한 개념인 '큐비트(qubit)'라는 최소 정보 단위를 사용하여 기존 컴퓨터에 비해 특정 문제를 더 빠른 속도로 해결할 수 있다.

혁명을 가져올 것으로 예상되는 양자 컴퓨터의 상용화를 위해 IBM, 구글을 비롯한 많은 글로벌 정보기술 기업들이 양자 컴퓨터 개발에 집중하고 있다. 양자 컴퓨터 개발의 선두 주자 중 한 곳인 구글은 2019년, 54-큐비트 양자 프로세서인 Sycamore로 기존의 슈퍼컴퓨터로 1만 년이 걸릴 것으로 예상되는 난수 증명 문제를 200초(3분 20초) 만에 해결하는 모습을 보이며 양자 컴퓨터가 기존 컴퓨터의 성능을 능가하는 것을 의미하는 '양자 우월성(Quantum supremacy)'을 최초로 달성하였다. 이어서 2023년, 구글은 슈퍼컴퓨터로 약 47년이 걸리는 작업을 6초 만에 해결하도록 Sycamore 프로세서를 발전시켜 양자 우월성 달성에 쐐기를 박았다. 현재 구글은 큐비트 수의 증가에 따라 양자 컴퓨터의 발생 빈도가 증가하는 문제에 집중하여, 오류 발생 확률이 기하급수적으로 증가하지 않게 코딩을 통해 보정해주는 오류 정정 기술 개발에 주력하고 있다. 또 다른 선두 주자인 IBM은 양자 프로세서 개발 로드맵을 공개하고 이를 차례차례 달성해 나가며 대규모 양자 컴퓨터의 개발에 앞장서고 있다. [표 1-1]은 IBM의 2026년까지 양자 프로세서 개발 로드맵을 정리한 것으로, 현재 IBM은 2023년 목표였던 1,121 큐비트 프로세서 Condor과 기존보다 오류율을 3배 낮춘 133 큐비트 프로세서 Heron 개발까지 완료한 상황이다.

[표 1-1] IBM의 양자 프로세서 개발 로드맵

Year	2019	2020	2021	2022	2023		2024		2025	2026
Processor	Falcon	Humming bird	Eagle	Osprey	Condor	Heron	Flamingo	Crossbill	Kookaburra	Scaling
Qubit	27	65	127	433	1,121	133	1,386	408	4,158	10K~ 100K

그러나 대규모 양자 컴퓨터의 개발과 Shor 알고리즘 및 Grover 검색 알고리즘 같은 양자 알고리즘의 등장은 기존의 암호화 체계의 안전성을 위협할 수 있다고 알려져 있다. Shor 알고리즘의 경우 공개키 암호의 기반 문제인 인수분해 문제를 다항 시간 안에 해결함으로써 공개키 암호 체계의 붕괴를 야기할 수 있다. 이러한 위협에 대비하기 위해 미국의 NIST(국립표준기술연구소)에서는 2017년부터 PQC(양자 내성 암호) 표준화 공모전을 개최하였고 2022년 표준암호와 후보 알고리즘을 발표하였다. 후보 알고리즘은 격자 기반 암호로 집중된 문제를 해결하기 위해 추가로 선정한 것이다. PQC 표준화 결과에 대한 내용은 [표 1-2]에서 확인할 수 있다.

[표 1-2] PQC 표준화 공모전 최종 결과

기능	알고리즘	기반문제	비고
공개키 암호화(PKE) /키캡슐화(KEM)	Classic McEliece	코드(Code) 기반	4라운드 후보 알고리즘
	BIKE		
	HQC		
	CRYSTALS-Kyber	격자(Lattice) 기반	표준화 알고리즘
전자서명 (Digital Signature)	CRYSTALS-Dilithium		
	FALCON		
	SPHINCS+	해시(Hash) 기반	

NIST는 선정한 후보 알고리즘이 모두 PKE/KEM이라는 문제를 극복하기 위해 전자서명 관련 공모전을 추가로 개최하였다. 현재 40개의 알고리즘이 제출되었고, 1라운드가 진행 중이다. 1라운드 후보 알고리즘에 대한 자세한 내용은 [표 1-3]에서 볼 수 있다.

[표 1-3] PQC 전자서명 표준화 공모전 1라운드 후보 알고리즘

기반문제	알고리즘	기반문제	알고리즘
코드 기반	CROSS	다변수(Multivariate) 기반	3WISE
	Enhanced pqsigRM		DME-Sign
	FuLeeca		HPPC
	LESS		MAYO
	MEDS		PROV
	Wave		QR-UOV
격자 기반	EagleSign		SNOVA
	EHTv3 and EHTv4		TUOV
	HAETAE		UOV
	HAWK		VOX
	HuFu	Symmetric 기반	AIMer
	Raccoon		Ascon-Sign
	SQUIRRELS		FAEST
MPC-in-the Head 기반	Biscuit		SPHINCS-alpha
	MIRA	아이소제니(Isogeny) 기반	SQIsign
	MiRitH	기타(Other)	ALTEQ
	MQOM		eMLE-Sig 2.0
	PERK		KAZ-SIGN
	RYDE		Preon
	SDitH		Xifrat1-Sign.I

이에 발맞춰 국내에서도 국산 양자 내성 암호를 개발하기 위해 양자내성암호연구단에서 KpqC 표준화 공모전을 진행하고 있다. 2023년 12월, 2라운드 진출자까지 공개가 되었으며 2라운드 후보와 관련된 정보는 [표 1-4]에서 볼 수 있다.

[표 1-4] KpqC 표준화 공모전 2라운드 후보 알고리즘

기능	알고리즘	기반문제
공개키 암호화/키캡슐화	NTRU+	격자 기반
	SMAUG+TiGER (merged)	
	PALOMA	코드 기반
	REDOG	
전자서명	AIMer	기타
	HAETAE	격자 기반
	NCC-Sign	
	MQ-Sign	다변수 기반

대칭키 암호의 경우 데이터 검색 복잡도를 낮출 수 있는 Grover 검색 알고리즘을 이용한 일반적인 공격(비밀키 전수 조사)에 의해 보안이 크게 손상될 수 있다. 대칭키 암호의 경우 기반 문제가 깨진 것은 아니기 때문에 다행히 공개키 암호처럼 아예 새로운 암호를 만들지 않고 일정한 보안 수준을 달성하는 기존의 암호를 사용할 수 있다. NIST에서는 양자 내성 암호 표준화를 위한 평가 기준을 안내한 문서에서 AES를 대칭키의 기준으로 잡고, 양자 공격 복잡도를 추정하는 기준을 제공하였다. 이에 최근 몇 년 동안 Grover 검색 알고리즘을 이용한 기존의 대칭키 암호들의 비밀키 복구에 발생하는 복잡도를 추정하고 해당 결과를 NIST에서 제시한 기준과 비교하여 양자 후 보안 강도를 평가하는 연구들이 발표되어 왔다. [표 1-5]는 국산 블록 암호에 대해 진행한 Grover 키 검색 비용 추정 연구 결과들을 모아놓은 것으로 가장 최신의 결과들만 가져온 것이다. 이 중 ARIA는 초경량 환경 및 하드웨어 구현에 최적화된 대칭키 암호 알고리즘으로 국내암호모듈검증제도(KCMVP)에 검증 대상으로 포함되어 있다. 이는 ARIA가 검증받은 암호모듈에 널리 사용되고 있다는 의미로, 다가올 위협에 대비하기 위해 ARIA의 양자 보안 강도를 측정하는 일은 상당히 중요하다. 다행스럽게도 이미 2020년에 ARIA의 양자 보안 강도를 측정한 연구가 존재한다. 그러나 [표 1-5]에 제시된 다른 블록 암호들은 모두 작년 내지는 올해 발표된 최적화 사례를 포함하여 이미 두세 번의 최적화가 진행되었지만, ARIA는 최적화된 연구가 한 번도 진행된 적이

없다. 또한, 2020년에 발표된 해당 연구는 큐비트 최적화에 초점이 맞춰져 있다. 최근 양자 컴퓨터 환경에서의 실행시간과 연관이 깊은 'depth'라는 지표의 최적화를 고려하는 연구가 증가하고 있다는 측면에서 ARIA에 대해 depth를 최적화한 연구를 진행할 필요가 있다.

[표 1-5] 국산 블록 암호에 대한 Grover 키 검색 비용 추정 연구 결과 비교 (recent result)

Cipher		Year	Qubit	Total gates	Total depth	Cost	NIST security
ARIA	128/128	2020	1,561	$1.998 \cdot 2^{85}$	$1.816 \cdot 2^{79}$	$1.814 \cdot 2^{165}$	Level 1
	128/192		3,121	$1.146 \cdot 2^{119}$	$1.073 \cdot 2^{112}$	$1.23 \cdot 2^{231}$	Level 3
	128/256		3,377	$1.384 \cdot 2^{151}$	$1.238 \cdot 2^{144}$	$1.714 \cdot 2^{295}$	Level 5
LEA	128/128	2022	389	$1.195 \cdot 2^{82}$	$1.247 \cdot 2^{77}$	$1.491 \cdot 2^{159}$	Level 1
	128/192		1,037	$1.775 \cdot 2^{115}$	$1.455 \cdot 2^{109}$	$1.292 \cdot 2^{225}$	Level 3
	128/256		1,165	$1.014 \cdot 2^{148}$	$1.645 \cdot 2^{141}$	$1.668 \cdot 2^{289}$	Level 5
HIGHT	64/128		457	$1.384 \cdot 2^{82}$	$1.901 \cdot 2^{75}$	$1.316 \cdot 2^{158}$	Level 1
CHAM	64/128	2023	391	$1.226 \cdot 2^{81}$	$1.002 \cdot 2^{76}$	$1.228 \cdot 2^{157}$	Level 1
	128/128		260	$1.305 \cdot 2^{81}$	$1.005 \cdot 2^{77}$	$1.311 \cdot 2^{158}$	Level 1
	128/256		775	$1.562 \cdot 2^{146}$	$1.198 \cdot 2^{141}$	$1.871 \cdot 2^{287}$	Level 5

NIST는 양자 내성 암호 표준화와 관련된 공식 문서에서 양자 컴퓨터가 실행할 수 있는 최대 회로 깊이를 의미하는 *MAXDEPTH* 라는 매개변수를 제안하였다. 양자 보안 강도를 평가하기 위해서는 양자 공격 복잡도뿐 아니라 실행과 관련된 *MAXDEPTH* 도 같이 고려해야 한다. 해당 내용에 대해서는 2장 4절과 4장 2절에서 보다 자세하게 설명한다.

본 논문의 구성은 다음과 같다. 2장에서는 논문의 배경으로 ARIA 블록 암호의 구성을 살펴보고 양자 컴퓨터에 대한 기초적인 정보를 다룬 후, 양자 회로의 구현 및 평가 방법에 대해 다룬다. 이어서 3장에서는 2장에 제시된 정보를 바탕으로 ARIA를 위한 양자 회로 설계를 제안한다. 4장에서는 양자 회에 대한 양자 자원 비용과 Grover 양자 회로에 대한 주요 검색 비용을 제시하고, 추정치를 기반으로 ARIA의 양자 보안 강도를 평가한다. 마지막으로

5장에서 결론과 향후 연구의 잠재적 방향에 관해 설명하며 논문을 마무리 짓는다.

1.2 연구기여

본 논문의 연구 기여는 다음과 같다.

첫 번째, ARIA의 depth를 최소화한 양자 구현이다. ARIA의 양자 회로 구현에서 우리의 주요 초점은 Toffoli-depth를 최소화하고 전체 depth를 보장하는 것이다. 최적화를 위한 다양한 기술을 통해 Toffoli-depth와 전체 depth의 감소를 달성하였다.

두 번째, 최적화를 위해 다양한 기술을 활용한다. depth를 감소시키기 위해 최적화를 위한 다양한 기술을 활용하였다. 이진 필드 연산을 최적화하기 위해 Karatsuba 알고리즘을 병렬로 구현하는 최적화된 곱셈과, 선형 행렬 최적화 기법을 사용한 제곱 방법을 선택하였다. 또한, 해당 구성 요소들에 대해 병렬 구현을 강화하였다.

세 번째, 양자 보안 강도를 평가한다. ARIA 양자 회로를 구현하는 데 필요한 자원을 추정하고 선행 연구와 비교하였다. 또한 구현된 양자 회로를 기반으로 Grover 키 검색 비용을 추정하고, 이를 NIST에서 제공하는 양자 보안 수준과 비교하여 ARIA의 양자 보안 강도를 평가하였다.

Ⅱ. 관련연구

2.1 블록암호 ARIA

ARIA는 Academy, Research Institute, Agency의 약자로 언급한 세 기관이 공동으로 개발한 국산 대칭키 블록암호이다. ARIA는 2004년 국가 표준 암호 알고리즘으로 채택된 이후, 보안 통신 및 데이터 보호를 위해 널리 사용되고 있다. 특히 ARIA는 KCMVP 검증 대상에 포함된 대칭키 암호로서 의의가 있다. ARIA는 설계자가 개발 과정에서 AES의 설계 원칙을 고려했기 때문에 대표적인 대칭키 암호인 AES와 유사한 인터페이스를 가지고 있고 마찬가지로 입출력 크기는 128-bit로 고정된 채, 키 크기만 128, 192, 256-bit로 달라진다. 그러나 AES와 달리 경량 하드웨어 구현에 최적화된 ISPN(Involutional Substitution-Permutation Network) 구조로 되어 있다는 특징을 갖는다.

2.1.1 라운드 함수

라운드 함수의 주요 작업은 라운드 키 덧셈(AddRoundKey), 치환 계층(SubstitutionLayer), 확산 계층(DiffusionLayer)까지 3단계로 구성된다.

라운드 키 덧셈 단계에서는 중간 상태 값을 각 라운드에 적합한 라운드 키와 XOR 연산한다.

치환 계층에서는 입력으로 들어온 128-bit 상태 값을 8-bit 단위로 나누고, S-box를 사용하여 치환을 수행한다. ARIA는 역 S-box를 포함해 4개의 S-box($S_1, S_1^{-1}, S_2, S_2^{-1}$)를 사용한다. S_1, S_1^{-1} 는 AES에서 사용되는 것과 동일하고 S_2, S_2^{-1} 는 ARIA를 위해 특별히 새로 설계된 S-box이다. ARIA에 사용되는 S-box들은 유한체 $GF(2^8)$ 상에서 x^{-1} 와 x^{247} 에 아핀 변화를 적용하여 생성된다. S-box S_1, S_2 는 8×8 정칙 행렬($\mathbf{A}$ 또는 $\mathbf{B}$)과 지수(x^{-1} 또는

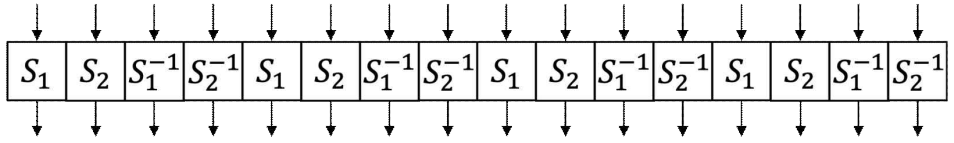
x^{247}) 사이의 곱을 수행하고 8×1 벡터와 XOR 연산을 진행한 후에 얻을 수 있다. 이 계산은 [수식 2-1]과 같이 표현될 수 있다.

$$S_1(x) = \mathbf{A} \cdot x^{-1} \oplus [1,1,0,0,0,1,1,0]^{\mathrm{T}},$$
$$S_2(x) = \mathbf{B} \cdot x^{247} \oplus [0,1,0,0,0,1,1,1]^{\mathrm{T}}$$

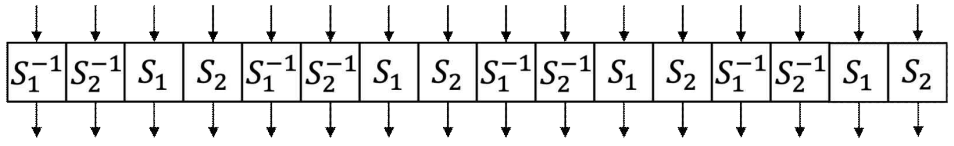
$$\text{where } \mathbf{A} = \begin{pmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 01011110 \\ 00111101 \\ 11010111 \\ 10011101 \\ 00101100 \\ 10000001 \\ 01011101 \\ 11010011 \end{pmatrix}$$

[수식 2-1] S-box 생성식

ARIA는 4개의 S-box로 구성된 두 가지 유형(유형1, 유형2)의 S-box 레이어를 가지고 있다. 유형1은 $S_1, S_2, S_1^{-1}, S_2^{-1}$ 순서로 구성된 32-bit 세트 4개로 구성된다. 두 유형은 서로 역관계이기 때문에 $(\text{유형1})^{-1} = (\text{유형2})$ 이다. 라운드 함수에서 유형1은 홀수 라운드에 사용되고 유형2는 짝수 라운드에 사용된다. [그림 2-1]은 ARIA의 유형별 S-box 레이어 구조를 그림으로 나타낸 것이다.



(*a*) 유형 1

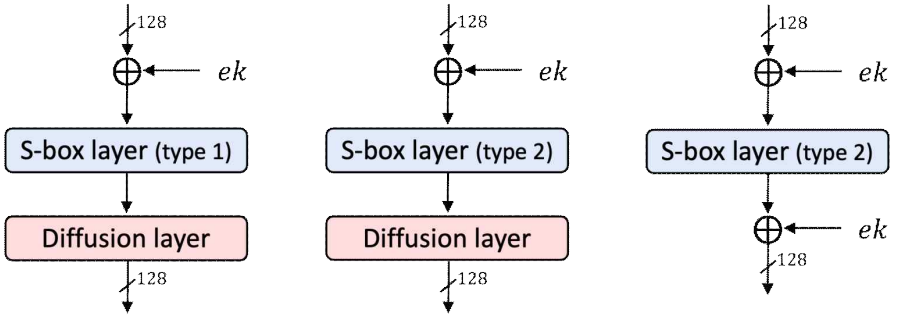


(*b*) 유형 2

[그림 2-1] 유형별 S-box 레이어 구조도

확산 계층에서는 주어진 16×16 involution 이진 행렬을 치환 계층의 출력과 곱하여 바이트 단위의 행렬 곱을 수행한다. Involution 이진 행렬은 역행렬이 기존 행렬과 동일하다는 특징을 가지고 있기 때문에 복호화 과정에서 별도의 역행렬을 구현할 필요가 없다.

라운드 함수의 세부 구성은 라운드가 홀수, 짝수, 최종인지에 따라 달라진다. 홀수 라운드와 짝수 라운드의 주요 차이점은 치환 계층에 사용된 S-box 레이어 유형으로, 홀수 라운드는 유형1을 사용하고 짝수 라운드는 유형2를 사용한다. 최종 라운드에서는 확산 계층이 생략되고 라운드 키 덧셈 단계가 한 번 더 수행된다. [그림 2-2]는 ARIA의 라운드 함수에 대한 개요를 그림으로 간단하게 나타낸 것이다.



$(a)\ F_o$ $(b)\ F_e$ $(c)\ F_f$

[그림 2-2] 라운드 함수 구조도

2.1.2 키 스케줄

키 스케줄은 키 초기화(KeyInit)와 키 생성(KeyGen) 단계로 구성된다. 우선, 키 초기화 단계는 라운드 키 생성을 위한 필수 구성 요소인 128-bit 초기 상수 W_0, W_1, W_2, W_3 을 생성한다. 해당 단계에는 라운드 함수 F_o, F_e 가 활용된다.

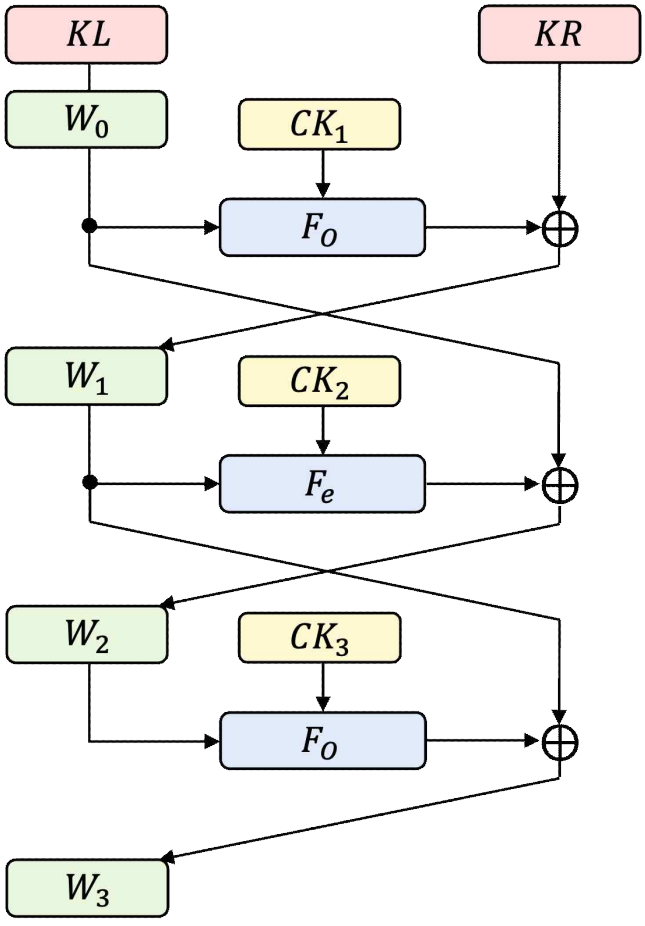
[수식 2-2]는 키 초기화 단계에서 입력 값 K_L, K_R 을 생성하는 데 사용되는 수식으로 마스터키 MK 에서 파생된다. 각각 128-bit인 K_L, K_R 을 연결한 결과

$(K_L \| K_R)$ 은 256-bit로 고정되기 때문에 MK 가 256-bit보다 작을 경우 부족한 비트를 0으로 채우는 패딩(padding)을 수행한다.

$$K_L \| K_R = MK \| 0 \ldots 0$$

[수식 2-2] K_L과 K_R 생성식

128-bit의 초기 라운드 상수 키 CK_1, CK_2, CK_3 은 π^{-1} 의 유리수 부분의 128-bit 상수 값으로, MK 의 길이에 따라 이를 사용하는 순서가 결정된다. [그림 2-3]은 키 초기화 단계 구조를 그림으로 나타낸 것이다.



[그림 2-3] 키 초기화 단계 구조도

키 생성 단계에서는 라운드 키가 생성되어 각 라운드의 키로 사용된다. 라운드 키 $ek_{1\sim17}$은 키 초기화 단계에서 생성된 128-bit 초기 상수 $W_{0\sim3}$에 회전 연산($\ll, \gg$) XOR 연산을 적용하면 얻을 수 있다.

ARIA의 라운드 키 크기는 모두 128-bit이고 ARIA-128, ARIA-192, ARIA-256의 라운드 수는 각각 12, 14, 16이다. 기본적으로 라운드 키의 개수는 라운드 수와 동일하지만, 최종 라운드에서 라운드 키 덧셈 작업이 두 번 수행되기 때문에 라운드 키가 1개 더 사용되어 ARIA-128, ARIA-192, ARIA-256에 대해 각각 13, 15, 17개의 라운드 키가 생성된다. 암호화에 사용되는 라운드 키 ek_i는 다음과 같이 생성된다.

$$
\begin{aligned}
ek_1 &= (W_0)\oplus(W_1 \gg 19), & ek_2 &= (W_1)\oplus(W_2 \gg 19)\\
ek_3 &= (W_2)\oplus(W_3 \gg 19), & ek_4 &= (W_0 \gg 19)\oplus(W_3)\\
ek_5 &= (W_0)\oplus(W_1 \gg 31), & ek_6 &= (W_1)\oplus(W_2 \gg 31)\\
ek_7 &= (W_2)\oplus(W_3 \gg 31), & ek_8 &= (W_0 \gg 31)\oplus(W_3)\\
ek_9 &= (W_0)\oplus(W_1 \ll 61), & ek_{10} &= (W_1)\oplus(W_2 \ll 61)\\
ek_{11} &= (W_2)\oplus(W_3 \ll 61), & ek_{12} &= (W_0 \ll 61)\oplus(W_3)\\
ek_{13} &= (W_0)\oplus(W_1 \ll 31), & ek_{14} &= (W_1)\oplus(W_2 \ll 31)\\
ek_{15} &= (W_2)\oplus(W_3 \ll 31), & ek_{16} &= (W_0 \ll 31)\oplus(W_3)\\
ek_{17} &= (W_0)\oplus(W_1 \ll 19)
\end{aligned}
$$

[수식 2-3] 암호화 라운드 키 ek_i

2.2 양자컴퓨터

2.2.1 큐비트

양자 컴퓨터에서 큐비트는 양자 역학의 중첩 성질을 띠기 때문에 0과 1이 확률로서 존재한다. 덕분에 하나의 큐비트로 0 혹은 1을 나타낼 수 있을 뿐 아니라, 0과 1이 동시에 존재하는 상태, 즉 중첩 상태도 표현할 수 있다. 수학적으로 표현하자면 단일 큐비트는 [수식 2-4]에 나타난 것처럼 서로 직교 관계인 2차원 열벡터 두 가지로 표현할 수 있으며, 양자 역학에서 양자 상태

를 표기할 때 사용하는 브라-켓(bra-ket) 표기법을 사용하면 오른쪽 꺽쇠괄호(〉)와 수직선(|)을 이용하여 각각 $|0\rangle$과 $|1\rangle$로 표기할 수 있다.

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix},\ |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

[수식 2-4] 단일 큐비트의 양자 상태 $|0\rangle$, $|1\rangle$의 벡터 표현

[수식 2-5]는 단일 큐비트의 중첩 상태 $|\psi\rangle$를 브라-켓 표기법으로 나타낸 것으로, 여기서 큐비트가 $|0\rangle$이 될 확률은 $|\alpha|^2$, $|1\rangle$이 될 확률은 $|\beta|^2$이다.

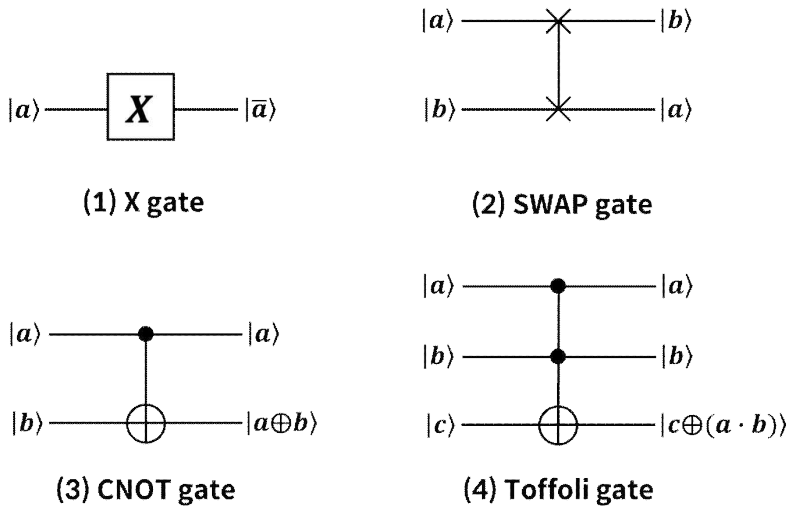
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$(\alpha, \beta \text{는 } |\alpha|^2 + |\beta|^2 = 1\text{을 만족하는 복소수})$$

[수식 2-5] 브라-켓 표기법으로 나타낸 중첩 상태 $|\psi\rangle$

또한, 큐비트는 양자 역학의 얽힘도 특징으로 갖기 때문에 큐비트 하나가 관측되어 상태가 확정될 경우 이와 얽혀 있는 다른 큐비트의 상태도 확정된다. 이러한 특징 덕분에 n개의 큐비트에 대한 모든 경우(2^n)의 연산을 동시에 수행하여 연산 속도를 증가시킬 수 있다.

2.2.2 양자 게이트

양자 컴퓨팅 환경에서는 AND, OR, XOR와 같은 디지털 논리 게이트를 그대로 사용하지 못하기 때문에 논리 게이트의 대체 수단으로 양자 게이트를 등장하였다. 본 절에서는 많은 양자 게이트 중 블록 암호 ARIA의 양자 회로 구현에 사용된 4가지 양자 게이트에 대해 중점적으로 설명한다. 다음 4가지 양자 게이트의 회로 구성은 [그림 2-4]에서 확인할 수 있다.



[그림 2-4] 양자 게이트 회로 구성

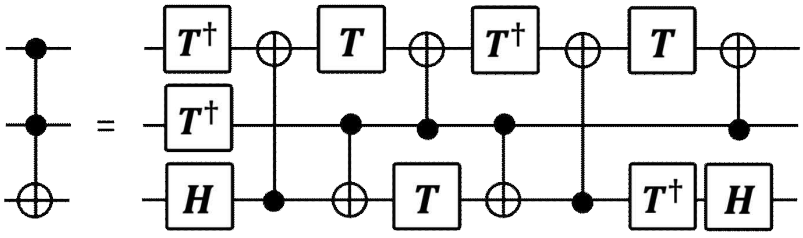
X 게이트는 기존 컴퓨터의 NOT 연산처럼 작동하여 큐비트의 상태를 반전시킨다. 예를 들어 0이 X 게이트의 입력으로 들어왔다면 출력은 1이 되는 것이다.

SWAP 게이트는 두 큐비트를 교환하는 역할을 한다. 예를 들어 SWAP(a, b)의 경우 a 위치에 b가, b 위치에 a가 들어가게 된다.

본 논문에서 가장 많이 사용되는 CNOT 게이트는 기존 컴퓨터의 XOR 연산처럼 동작한다. CNOT 게이트의 입력으로 들어오는 두 큐비트는 각각 제어 큐비트(control qubit), 대상 큐비트(target qubit) 칭하는데, 제어 큐비트의 상태에 따라 대상 큐비트의 상태가 결정된다. 예를 들어 CNOT(a, b)에서 만약 제어 큐비트 a의 상태가 1이면, 대상 큐비트 b의 상태가 뒤집힌다. 결과적으로 제어 큐비트 a의 상태 변화 없이, 대상 큐비트 b에 a⊕b의 값이 저장된다(즉, b = a⊕b).

마지막으로 Toffoli(a, b, c)로 표현되는 Toffoli 게이트는 기존 컴퓨터에서 AND 연산처럼 작동한다. 3개의 입력 큐비트를 필요로 하며 처음 2개의 큐비트 a, b는 제어 큐비트 역할을 하고 나머지 큐비트 c는 대상 큐비트 역할을 한다. 두 제어 큐비트가 모두 상태 1인 경우에만 대상 큐비트 c의 상태가 반전된다. 큐비트 a와 b의 상태는 유지한 채, a & b 작업의 결과가 큐비트 c

와 XOR되는 것이다(즉, c = c ⊕ (a & b)). 앞서 다룬 3개의 게이트들과 달리 Toffoli 게이트는 조금 더 복잡한 구성을 갖는다. 일반적으로 Toffoli 게이트는 7개의 T 게이트와 T 게이트 외의 게이트들을 의미하는 Clifford 게이트 8개로 구성된다. 표준 Toffoli 게이트는 T-count(T 게이트의 개수)가 7이고, T-depth(T 게이트의 깊이)는 6이다. 4개의 게이트 중 가장 높은 양자 비용을 차지하는 Toffoli 게이트만 최적화해도 회로의 최적화를 상당 부분 달성할 수 있기 때문에, depth와 T-depth를 최소화한 Toffoli 게이트 회로 구현 관련 연구가 지속적으로 보고되고 있다. 본 논문에서는 많은 연구 중 Amy 등이 제안한 Toffoli 게이트 구현을 사용하였다. [그림 2-5]는 Amy가 제안한 Toffoli 게이트의 회로 구성으로, T-count는 표준으로 제시된 것과 동일하게 7이고, T-depth는 4, Full-depth는 8을 갖는다.



[그림 2-5] Toffoli 게이트 회로 구성

2.2.3 양자 회로

양자 회로(Quantum Circuit)는 기존 컴퓨터의 논리 회로와 유사한 것으로, 시간 흐름에 따른 큐비트의 양자 상태 변화를 나타낼 수 있다. 양자 회로에서 depth는 연속적인 양자 게이트의 수를 의미하며, 양자 컴퓨터에서의 연산 시간과 연관되어 있다. 양자 회로에서 와이어의 개수는 물리적인 큐비트의 개수와 직결된다.

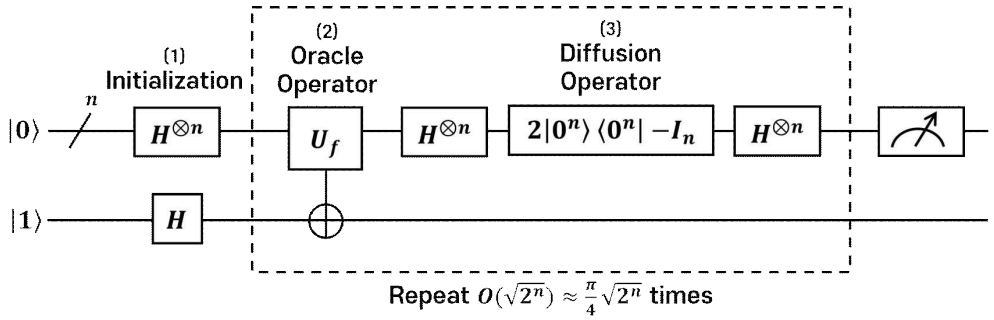
현재 개발이 진행되고 있는 양자 컴퓨터는 기존 컴퓨터보다 자원의 제약이 매우 크다. 따라서, 반드시 양자 회로의 최적화를 통해 사용되는 양자 자원(큐비트 수, depth, 양자 게이트 수 등)을 최소화해야 한다. 양자 회로를 최

적화하는 일반적인 방법으로는 큐비트 최적화와 depth 최적화가 있으며, 전자는 기존에 사용한 연산을 한 번 더 적용하는 리버스 연산(reverse operation)을 통해 큐비트를 초기화하여 다른 연산에 재활용함으로써 큐비트를 절약한다. 이는 리버스 연산이 필수적이기 때문에 depth 증가가 불가피하다는 단점이 존재한다. 후자는 리버스 연산 대신 보조 큐비트를 의미하는 ancilla qubit를 추가로 사용하여 병렬 연산을 구성함으로써 depth를 감소시킨다. 이 또한 마찬가지로 ancilla qubit를 사용하기 때문에 전체 큐비트 수가 증가하게 된다는 단점이 존재한다. 이를 통해서 depth와 큐비트는 일종의 trade-off 관계라 볼 수 있다.

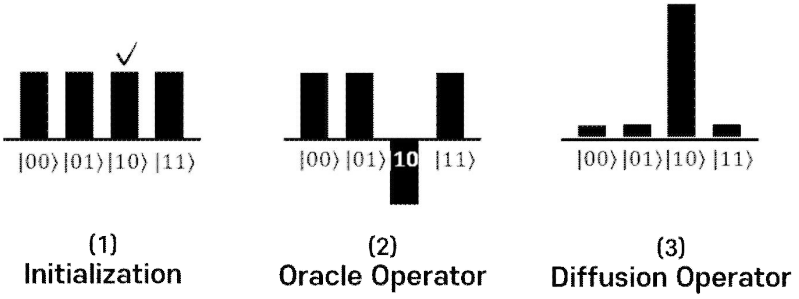
2.3 Grover 키 검색 알고리즘

1996년 Lov Grover에 의해 제안된 Grover 검색 알고리즘은 구조화되지 않은 데이터베이스 집합 N에서 특정 데이터를 빠르게 검색할 수 있게 해주는 양자 알고리즘으로, 검색 복잡도를 $O(N)$에서 $O(\sqrt{N})$으로 감소 시켜준다. 이를 대칭키 암호의 n-bit 비밀키 검색에 적용할 경우, 무차별 대입 공격(brute-force attack)의 검색 복잡도를 $O(2^n)$에서 $O(2^{n/2})$로 줄여 이론적으로 보안 수준이 절반으로 줄어든다.

Grover 키 검색 알고리즘은 초기화(Initialization), 오라클 연산자(Oracle Operator), 확산 연산자(Diffusion Operator)로 구성된다. 3가지 단계로 구성된 Grover 검색 알고리즘의 양자 회로는 [그림 2-6]에서 확인할 수 있다. [그림 2-7]은 이해를 돕기 위하여 n이 2일 때, 해답을 $|10\rangle$이라 가정한 경우 Grover 검색 알고리즘 작동 과정을 시각적으로 표현한 것이다.



[그림 2-6] Grover 검색 알고리즘의 양자 회로



[그림 2-7] Grover 검색 알고리즘 작동 과정 ($n=2$)

(1) 초기화 단계 : 초기화 단계에서는 n-qubit의 키를 Hadamard 게이트의 입력으로 넣어 2^n 개의 상태가 모두 동일한 진폭을 갖는 중첩 상태 $|\psi\rangle$를 생성한다.

$$|\psi\rangle = (H|0\rangle)^{\otimes n} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

[수식 2-6] 중첩 상태 $|\psi\rangle$

이는 [그림 2-7]의 첫 번째 그래프에 해당한다. Hadamard 게이트를 통과한 결과를 나타내며, 4개의 상태($|00\rangle, |01\rangle, |10\rangle, |11\rangle$)가 모두 동일한 진폭을 갖는 것을 확인할 수 있다.

(2) 오라클 연산자 : 대상 암호에 대한 양자 회로는 오라클에서 중첩 상태를 통해 생성된 키(준비된 키)를 사용하여 알려진 평문을 암호화하고 모든 키 값에 대한 암호문을 생성한다. 오라클 연산자 U_f에서 사용된 함수 $f(x)$는 생성된 암호문 $Enc_{key}(p)$을 알려진 암호문 c와 비교하여 일치하지 않으면 0을 반환하고 일치하면 1을 반환한다. 만약, 두 암호문을 비교한 결과 일치하는 항목이 나올 경우, [수식 2-7]에서 $f(x)=1$이므로 해당 키 상태의 진폭은 음수가 된다. 일치하는 항목이 없는 경우 $(-1)^0=1$이기 때문에 진폭은 양수로 유지된다.

$$f(x)=\begin{cases}1 \text{ if } Enc_{key}(p)=c\\ 0 \text{ if } Enc_{key}(p)\neq c\end{cases}$$

$$U_f(|\psi\rangle|-\rangle=\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle|-\rangle$$

[수식 2-6] 오라클 연산자 U_f

오라클 연산자를 통과하여 해답인 상태의 진폭이 음수로 바뀌는 것은 [그림 2-7]의 두 번째 그래프에서 볼 수 있다.

(3) 확산 연산자 : 확산 연산자 D는 음의 진폭을 갖는 키 상태 (대상키의 상태)를 대칭 상태로 변환하는 역할을 한다. 변환은 모든 키 상태 진폭의 평균을 계산한 후, 각 키 상태(I_n)의 진폭에서 평균값을 빼는 과정으로 구성되어 있다. 만약, 오라클 연산을 거친 후 진폭으로 음수를 가질 경우, 음수에서 음수를 빼는 것이므로 양수가 되어 해당 값의 진폭만 증폭되고, 나머지 상태들은 모두 진폭이 줄어든다.

$$D=2|s\rangle\langle s|-I_n$$

[수식 2-7] 확산 연산자 D

확산 연산을 수행하고 난 후, 음수의 진폭을 가지던 해답 $|10\rangle$의 진폭이

갑자기 증가하는 모습은 [그림 2-7]의 세 번째 그래프에서 확인할 수 있다.

솔루션 키를 측정할 확률을 높이기 위해서 (2) 오라클 연산자와 (3) 확산 연산자를 충분히 반복해야 한다. 일반적으로 반복횟수가 $\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ 일 때 정답을 측정할 확률이 가장 높은 것으로 알려져 있다. $\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ 만큼 반복을 수행한다면 n이 2보다 큰 경우에도 [그림 2-7]의 마지막 그래프처럼 명확하게 나올 수 있는 것이다.

2.4 NIST의 양자 보안 강도와 MAXDEPTH

NIST는 양자 내성 암호(Post-quantum cryptography) 표준화 공모전 진행 중에 제출된 알고리즘들을 분석하기 위해 관련 공식 문서에서 기존 NIST 대칭키 암호 표준에 명시된 보안 강도 범위를 기반으로 보안 기준을 제공하였다. 양자 내성 보안 기준은 AES와 SHA-2/3에 대한 양자 공격의 복잡도를 기반으로 하며 블록 암호의 경우 AES의 보안 기준을 참조한다.

Grover 검색 알고리즘은 대칭키 암호를 표적으로 삼는 가장 효율적인 양자 공격 중 하나로 인식되며 NIST도 이 사실을 인정하였다. 5단계의 보안 레벨 중 레벨 1, 3, 5는 각각 AES-128, AES-192, AES-256에 대한 Grover 키 검색에 필요로 하는 비용에 따라 평가된다. 이 비용은 총 게이트 수에 Grover 검색 회로의 depth를 곱한 값이다. 과거 몇 년간 발표된 연구를 통해 AES의 양자 회로를 최적화하여 Grover 키 검색 비용을 줄인 결과, NIST는 최근 공식 문서에서 레벨 1, 3, 5의 비용을 각각 $2^{157}, 2^{221}, 2^{285}$로 정의했다. 이는 Jaques et al. 이 Eurocrypt'20에서 제시한 AES의 대해 깊이 최적화된 양자 회로 구현의 비용을 인용한 것이다.

그러나, Jaques 의 양자 회로 구현에는 몇 가지 프로그래밍 관련 문제가 있다. Jang et al. 은 논문에서 이러한 문제들을 해결하고 조사하여 논문에서 제시한 최적화된 AES 양자 회로를 통해 Jaques 의 논문에서 언급된 비용을 대략 달성할 수 있음을 보였다. 현재까지 이해하기로는 Jang 의 논문에 제시

된 AES의 양자 회로 구현 비용 $2^{157}, 2^{192}, 2^{274}$ 이 AES 양자 회로 최적화에 대해 가장 주목할 만한 결과이다.

추가적으로 *MAXDEPTH* 라는 접근 방식도 고려해야 한다. NIST는 양자 컴퓨터가 실행할 수 있는 최대 회로 깊이를 나타내는 *MAXDEPTH* 라는 매개변수를 도입하였다. 지나치게 depth가 크면 시간 측면에서 실행 문제가 발생할 수 있기 때문이다. NIST에서 제공하는 양자 공격에 대한 깊이 제한(*MAXDEPTH*)은 $2^{40}, 2^{64}, 2^{96}$ 을 초과하지 않는 것이다.

Ⅲ. 제안 기법

본 장에서는 ARIA에 대해 최적화된 양자 회로 구현을 제시한다. ARIA를 양자 회로로 구현한 선행 연구의 결과와 비교하며 최적화된 구성 요소들을 설명한다. 최적화된 구성 요소는 3가지로 S-box, 확산 계층, 그리고 키 스케줄이 있다. 1절에서는 치환 계층에서 사용되는 S-box의 양자 회로에 관해 서술한다. S-box 양자 회로의 최적화를 위하여 역연산에서 가장 큰 비용을 차지하는 곱셈 연산에 사용된 알고리즘을 바꾸었고, 제곱 연산에 사용된 선형 행렬 최적화 방법을 바꾸었다. 2절에서는 확산 계층의 양자 회로에 관해 서술한다. 확산 계층 또한 선형 행렬 연산에 해당하기 때문에 1절의 제곱 연산에 사용된 방법을 동일하게 적용하였다. 마지막으로 3절에서는 키 스케줄의 양자 회로 구현에 관해 서술한다. 키 스케줄에서는 기본적으로 상수가 많이 사용되기 때문에, 상수로 정의된 변수를 양자 회로에 적용할 때 갖는 특징들을 활용하는 방법들을 제시 및 적용하였다.

3.1 S-box 양자 회로 구현

기존 컴퓨터에서 AES를 포함한 대다수의 블록암호의 S-box는 사전에 치환할 값들이 정의되어 있는 look-up table 방식을 사용한다. 그러나 양자 컴퓨팅 환경에서는 관측하기 전까지는 큐비트의 상태를 알 수 없고, 큐비트 수 또한 제한되어 있기 때문에 기존의 look-up table 방식을 사용하는 것은 바람직하지 못하다. 따라서 S-box의 생성식을 기반으로 양자 회로를 구현하는 일은 불가피하다. 물론, 기존의 S-box 치환표를 기반으로 양자 회로를 효율적으로 구성해 주는 LIGHTER-R이라는 도구를 사용하면 간단하게 S-box의 양자 회로를 구할 수 있다. 그러나 해당 도구는 4-bit S-box에만 적용할 수 있기 때문에 8-bit S-box를 사용하는 ARIA에는 적용할 수 없다. 따라서 ARIA의 S-box의 양자 회로를 구현하기 위해서는 곱셈 역수를 구하고 아핀 변환을 수행할 필요가 있다.

3.1.1 유한체 $GF(2^8)$ 상에서의 변환

ARIA의 S-box S_1, S_2 를 구하기 위해서는 2.1.1절의 [수식 2-1]에 나온 x^{-1} 와 x^{247} 를 구해야 한다. S_2 의 x^{247} 은 유한체 $GF(2^8)$ 상에서 기약 다항식 $m(x)$ 을 사용하여 [수식 3-1]과 같이 x^{-1} 을 활용한 식으로 표현할 수 있다. x 의 역원(x^{-1})을 구하는 것이 가장 큰 핵심이 되는 것이다.

$$x^{247} \equiv (x^{-1})^8 \equiv (((x^{-1})^2)^2)^2 \bmod m(x)$$
$$m(x) = x^8 + x^4 + x^3 + x + 1$$

[수식 3-1] $GF(2^8)$ 상에서의 x^{247} 변환

마찬가지의 방법으로 $GF(2^8)$ 상에서 x^{-1} 은 x^{254} 으로 표현이 가능하다. x^{254} 는 Itoh-Tsujii 알고리즘을 통해 제곱과 곱셈으로 구성된 수식으로 바꿈으로써 보다 효율적으로 구할 수 있다. x^{254} 에 Itoh-Tsuji Inversion 알고리즘을 적용한 결과는 [수식 3-2]와 같다.

$$x^{-1} = (x^{254}) = ((x \cdot x^2) \cdot (x \cdot x^2)^4 \cdot (x \cdot x^2)^{16} \cdot x^{64})^2$$

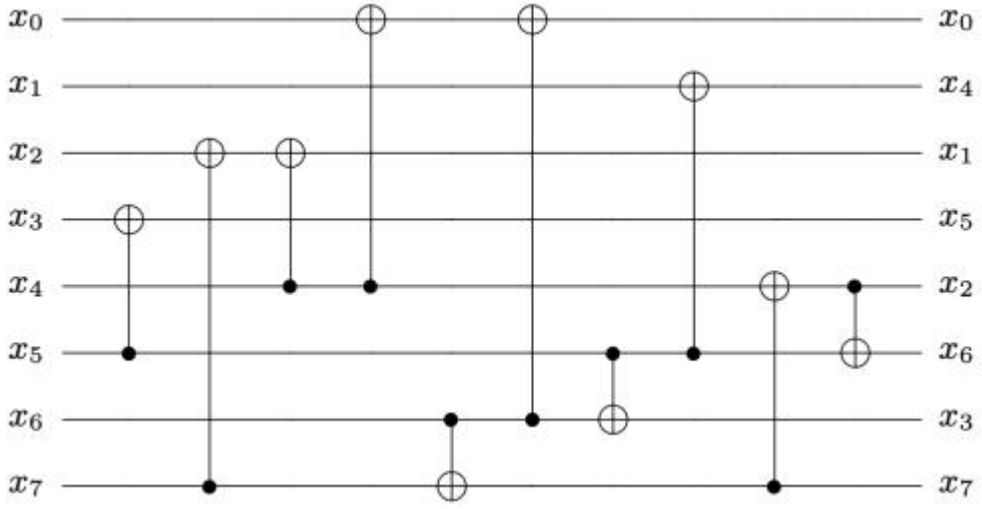
[수식 3-2] x^{254} 에 대한 Itoh-Tsuji Inversion 알고리즘 적용 결과

3.1.2 제곱 연산

유한체 $GF(2^8)$ 상에서의 제곱 연산은 일반적으로 다항식을 자기 자신과 곱셈하여 새로운 다항식을 얻고, 유한체의 크기에 맞추기 위해 기약 다항식으로 나누는 모듈러 축소를 진행하는 단계로 구성된다. 보통 다항식 곱셈보다 행렬 곱셈이 더 빠르기 때문에, 연산 속도를 높이기 위해서 제곱 연산은 기약 다항식 기반의 제곱 행렬을 활용한 행렬 연산 형태로 변환하여 수행하였다. 행렬 곱셈은 XOR 연산을 사용하기 때문에 행렬 연산 형태로 제곱 연산을

진행할 경우 XOR 연산으로만 구성된다. 여기에 Xiang et al. 이 제안한 이진 행렬의 인수분해 기반의 휴리스틱 검색 알고리즘인 XZLBZ를 이용하면 XOR 연산만을 이용한 in-place 구조로 구현할 수 있다.

ARIA의 제곱 연산은 모듈러 축소 및 XZLBZ를 통해 CNOT 게이트와 SWAP 게이트를 사용하여 구현되었다. 제곱 연산의 양자 회로를 나타낸 [그림 3-1]을 보면 10개의 CNOT 게이트를 사용하고 depth가 7인 것을 알 수 있다.



[그림 3-1] 제곱 연산의 양자 회로

3.1.3 곱셈 연산

곱셈 연산에서는 WISA'22에서 처음 발표된 Jang et al. 의 Toffoli-depth 최적화 Karatsuba 양자 곱셈을 사용한다. 곱셈 횟수를 줄이는 것으로 알려진 Karatsuba 알고리즘을 사용하면 곱셈 연산에 필요로 하는 Toffoli 게이트의 수를 감소시킬 수 있다. Jang et al.의 곱셈은 이러한 특징을 갖는 Karatsuba 알고리즘을 재귀적으로 적용하여 모든 곱셈, 즉 AND 연산을 독립적으로 수행한다. AND 연산을 독립적으로 수행할 수 있기 때문에 Toffoli 게이트를

병렬로 구성함으로써 Toffoli-depth를 상당 부분 감소시킬 수 있다. 여기에 해당 양자 곱셈은 추가적으로 많은 보조 큐비트를 할당시킴으로써 Toffoli-depth 1을 달성하였다. 이 곱셈 연산은 양자-양자 값 간의 곱에만 사용된다.

ARIA의 선행 연구인 Chauhan et al.의 양자 구현에서는 Schoolbook 곱셈 방법을 사용했지만, 우리의 작업에서는 Jang et al.의 Toffoli-depth 최적화된 Karatsuba 곱셈 방법을 채택함으로써 양자 자원을 크게 절감하였다.
[표 3-1]은 각 방법의 곱셈 연산에서 필요로 하는 양자 자원을 비교한 결과를 표로 정리한 것이다. [표 3-1]을 보면 이전 연구에서 사용한 방법보다 전반적으로 양자 자원이 감소하였는데, 특히 Toffoli-depth가 상당 부분 최적화되었음을 알 수 있다.

[표 3-1] 곱셈 연산 양자 자원 비교

Source	#Clifford	#T	Toffoli-depth	Full-depth
Schoolbook (Cheung et al.)	435	448	28	195
Toffoli-depth에 최적화된 Karatsuba (Jang et al.)	390	189	1	28

※: The multiplication size n is 8.

3.1.4 행렬-벡터 곱과 벡터 XOR 연산

제곱, 곱셈 연산을 통해 x^{-1}와 x^{247}를 구한 후, 8×8 정칙 행렬($\mathbf{A}$ 또는 $\mathbf{B}$)과 지수(x^{-1} 또는 x^{247}) 사이의 행렬-벡터 곱은 고전 값과 양자 값 사이의 곱이기 때문에 3.1.2에서 사용한 XZLBZ 방법을 적용하여 최적화하였다. 8×1 벡터와의 XOR 연산은 원래 8×1 벡터를 제어 큐비트로 두고 상태 큐비트를 대상 큐비트로 두어서 8-qubit에 모두 CNOT 게이트를 적용해야 한다. 그러나 주어진 벡터가 상수라는 점을 활용하면 상수 벡터의 요소가 1인 위치만 상태 큐비트의 반전이 일어나기 때문에, 8개의 큐비트에 모두 CNOT

게이트를 적용할 필요 없이 반전이 필요한 위치에만 X 게이트를 적용하는 형식으로 바꿀 수 있다. 이를 통해 사용되는 양자 게이트의 수를 줄였고 CNOT 게이트보다 훨씬 간단한 X 게이트를 사용함으로써 양자 자원을 절약하였다.

3.2 확산 계층의 양자 회로 구현

ARIA의 확산 함수 $A : GF(2^8)^{16} \rightarrow GF(2^8)^{16}$ 은 16×16 이진 행렬 곱으로 표현된다. 이진 행렬의 한 요소는 바이트 단위이기 때문에 입력으로 들어온 큐비트와 곱하기 위해서는 바이트를 비트 단위로 변환한 후에 연산을 수행해야 한다. 이를 위해 행렬의 원소 0은 8×8 제로 행렬을 의미하고, 행렬의 원소 1은 8×8 항등 행렬을 의미한다고 가정하고 계산을 진행하였다. 여기까지는 선행 연구인 Chauhan et al. 과 동일하다. 차이점은 행렬-벡터 곱셈을 구현에 적용된 선형 행렬 최적화 방법이다. Chauhan 은 PLU 분해(PLU factorization)를 사용한 반면, 우리의 논문에서는 앞서 사용했던 XZLBZ를 적용하여 더욱 회로의 효율성을 증가시켰다. [표 3-2]는 확산 레이어에 사용된 양자 자원을 비교한 결과이다. Chauhan 은 바이트 당 96개의 CNOT 작업을 필요로 하기 때문에 총 $768 (= 96 \times 8)$ 개의 CNOT 게이트가 사용되었다. 반면에 제안하는 양자 구현에서는 바이트 당 47개의 CNOT 작업을 필요로 하기 때문에 총 $376 (= 47 \times 8)$ 개의 CNOT 게이트가 사용되었다. 결론적으로 큐비트 수는 그대로 유지하면서 CNOT 게이트는 51.04%, depth는 45.16% 감소되었다.

[표 3-2] 확산 계층 양자 자원 비교

Method	#CNOT	#qubit	Depth
PLU factorization	768	128	31
XZLBZ (This work)	376		17

3.3 키 스케줄의 양자 회로 구현

키 스케줄 단계에서는 상수로 정의된 변수를 활용하는 부분의 최적화에 집중함으로써 큐비트와 양자 게이트를 비롯한 양자 자원을 감소시켰다. [알고리즘 3-1]은 키 스케줄 단계의 양자 회로 구현을 알고리즘으로 표현한 것이다. 6번째 줄까지가 키 초기화 단계이고, 그 이후부터가 라운드 키 생성 단계이다.

Algorithm 1 Quantum circuit implementation of key schedule for ARIA

Input: master key MK, key length l, vector a, b, ancilla qubit anc, round number r

Output: round key ek

▷ Key Initialization

1: $W_1 \leftarrow F_o(MK[:128], a, b, anc)$ ▷ $MK[:128]$ is K_L

2: Constant_XOR(W_1 $[l-128:128]$, $MK[l-128:l]$) ▷ $MK[l-128:l]$ is K_R

3: $W_2 \leftarrow F_e(W_1, a, b, anc)$

4: $W_2 \leftarrow$ CNOT128($MK[:128]$, W_2)

5: $W_3 \leftarrow F_o(W_2, a, b, anc)$

6: $W_3 \leftarrow$ CNOT128(W_1, W_3)

▷ Key Generation

7: num = [19, 31, 67, 97, 109]

8: **for** $i \leftarrow 0$ to r **do**

9: **Transform** ek:

10: **if** $i = 0 \pmod 4$ **then**

11: Constant_XOR(ek, $MK[:128]$)

12: **else**

13: $ek \leftarrow$ CNOT128($W_{(i\%4)}$, ek)

14: $ek \leftarrow$ CNOT128($W_{((i+1)\%4)} \ggg$ num$[i\%4]$, ek)

15: **return** ek

16: Use ek in $i-th$ round function

17: **if** i is not r **then**

18: **Reverse** (**transform** ek)

19: **return** ek

20: **end for**

[알고리즘 3-1] ARIA 키 스케줄

3.3.1 키 초기화 단계

키 초기화 단계에서는 라운드 함수를 사용하여 128 큐비트 W_1, W_2, W_3이 생성된다. 키 스케줄에 대한 K_L은 W_0의 생성에만 사용되기 때문에 새로운 큐비트로 W_0를 할당하는 대신 K_L을 사용하여 128 큐비트를 절약할 수 있다. 또한, K_R과 W_1의 XOR 연산을 수행할 때, K_R은 상수이기 때문에 K_R의 비트가 1일 때에만 W_1의 해당 위치에 X 게이트가 적용된다. CNOT 게이트를 더 저렴한 X 게이트로 대체함으로써 게이트 비용을 감소시킬 뿐 아니라 필요로 하는 위치에만 양자 게이트를 적용함으로써 사용된 게이트 수도 감소시켰다. K_R, W_1의 XOR 연산과 관련된 내용은 [알고리즘 3-1]의 2번째 줄에서 확인할 수 있다.

3.3.2 키 생성 단계

키 생성 단계에서는 $W_{0\sim3}$을 이용하여 각 라운드의 암호화키로 사용되는 라운드 키 ek를 생성한다. ek 생성에 W_0을 사용하는 경우는 3.3.1에서 사용한 두 가지 방법을 같이 적용하여 최적화할 수 있다. W_0를 K_L로 대체하고, K_R과 W_1의 XOR 연산에 사용한 것과 마찬가지로 CNOT 게이트를 X 게이트로 대체하는 방법을 적용하여 게이트 비용을 줄였다. 해당 내용은 [알고리즘 3-1]의 11번째 줄에서 볼 수 있다.

이론적으로는 ek의 값이 라운드마다 다르기 때문에 매번 새로운 큐비트를 할당하고 값을 저장해야 한다. 큐비트를 재사용하기 위해서는 실행한 연산을 거꾸로 다시 수행함으로써 큐비트를 초기화하면 된다. 일반적으로는 리버스 연산이 depth 증가와 관련이 깊기 때문에 depth 최적화를 위해서는 리버스 연산을 최소화하는 것이 좋다. 그러나 이전 라운드의 결과가 다음 라운드에 영향을 미치는 구조라 병렬화가 힘들고, X 게이트 및 CNOT 게이트로 구성된 키 생성 단계에 대한 리버스 연산이 depth 증가에 큰 영향을 미치지 않기

때문에 라운드마다 128개의 보조 큐비트를 추가로 할당하는 것보다는, 리버스 연산을 수행하여 초기화한 큐비트를 재활용하는 것이 훨씬 효율적이다. 이러한 이유에서 매 라운드마다 ek에 대한 새 큐비트를 생성하고 할당하는 대신, 매 라운드가 끝날 때 라운드 키 생성에 대한 리버스 연산을 수행하여 ek를 초기화하고 다음 라운드에서 재사용하였다. 해당 내용은 [알고리즘 3-1]의 18번째 줄에서 확인할 수 있으며, 이때 9~15번째 줄의 연산에 대해서 리버스 연산을 수행한다.

Ⅳ. 평가

본 장에서는 ARIA에 대한 양자 회로 자원을 추정하고 분석한다. 제안된 양자 회로는 아직 대규모 양자 컴퓨터에서 구현할 수 없다. 따라서 실제 양자 컴퓨터 대신 기존 컴퓨터 환경에서 IBM에서 제공하는 양자 프로그래밍 도구인 ProjectQ를 사용하여 양자 회로를 구현하고 시뮬레이션하였다. ProjectQ에서 제공하는 내장 라이브러리인 ClassicalSimulator에서는 2.2.2절에서 설명한 4가지 양자 게이트로만 이뤄진 제한된 양자 회로를 시뮬레이션할 수 있다. 이 경우 많은 수의 큐비트를 사용해 볼 수 있다. ClassicalSimulator를 사용할 경우, 특정 입력에 대한 출력을 고전적으로 계산하여 확인해 봄으로써 양자 회로 구현이 제대로 이뤄졌는지 테스트해 볼 수 있다. 양자 자원을 추정해 보기 위해서는 ResourceCounter라는 또 다른 내부 라이브러리가 필요하다. ResourceCounter는 큐비트 개수, 양자 게이트 개수, 회로의 depth만을 계산하여 출력한다.

4.1 성능 평가

[표 4-1]과 [표 4-2]는 ARIA의 양자 회로 구현에 필요로 하는 양자 자원을 나타낸 것이다. 해당 표 모두 Chauhan 에 의해 제안된 양자 회로와 본 논문에서 제안한 양자 회로 간의 양자 자원을 비교한다. 표에서 CS는 Chauhan 의 양자 회로를 의미한다. [표 4-1]은 ARIA의 NCT (NOT, CNOT, Toffoli) 게이트 수준에서의 양자 자원을 보여주며, [표 4-2]는 Toffoli 게이트를 분해하여 얻은 Clifford+T 수준에서의 양자 자원을 제시한다. Chauhan 의 논문에서는 Clifford+T 수준에서의 양자 자원이 명시적으로 제공되지 않았기 때문에, [표 4-2]의 양자 자원은 Chauhan 의 논문에서 제공된 정보를 기반으로 추정하였다. 더불어, 본 논문에서 제안하는 구현은 큐비트 활용도와의 균형을 고려하면서 회로의 depth 최적화에 중점을 둔 것이다. 본 장에서 $TD-M$ 비용을 고려함으로써 회로 복잡도 지표를 포괄하는

평가를 수행하였다.

[표 4-1] ARIA 양자 회로에 대한 양자 자원 추정 결과 비교 (NCT Level)

Cipher	Source	#X	#CNOT	#Toffoli	Toffoli-depth	#qubit	Depth	$TD-M$
ARIA-128	CS	1,595	231,124	157,696	4,312	1,560	9,260	6,726,720
	This work	1,408	272,392	25,920	60	29,216	3,091	1,752,960
ARIA-192	CS	1,851	273,264	183,368	5,096	1,560	10,948	7,949,760
	This work	1,624	315,144	29,376	68	32,928	3,776	2,239,104
ARIA-256	CS	2,171	325,352	222,208	6,076	1,688	13,054	10,256,288
	This work	1,856	352,408	32,832	76	36,640	4,229	2,784,640

[표 4-2] ARIA 양자 회로에 대한 양자 자원 추정 결과 비교 (Clifford+T Level)

Cipher	Source	#Clifford	#T	T-depth	#qubit	Full-depth
ARIA-128	CS	1,494,287	1,103,872	17,248	1,560	37,882
	This work	481,160	181,440	240	29,216	4,241
ARIA-192	CS	1,742,059	1,283,576	20,376	1,560	44,774
	This work	551,776	205,632	272	32,928	5,083
ARIA-256	CS	2,105,187	1,555,456	24,304	1,688	51,666
	This work	616,920	229,824	304	36,640	5,693

4.2 Grover 키 검색 비용 평가

본 절에서는 ARIA의 양자 보안 강도를 평가하고 해당 암호 알고리즘에 대한 Grover 키 검색 비용을 추정한다. 확산 연산자의 오버헤드는 오라클 연산자의 오버헤드와 비교하면 무시할 수 있을 만큼 미미하다고 간주되기 때문에 Grover 키 검색 비용을 추정할 때 확산 연산자의 오버헤드는 무시된다.

따라서, Grover 키 검색 비용 추정 시에 오라클 연산자의 수행만 고려한다. n-비트 비밀키를 사용하는 대칭키 암호에 대한 Grover 키 검색을 성공하기 위해서는 2.3절에 설명한대로 대략 $\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ 번 반복해야 한다.

2.4절에서 언급한 Jaques 의 논문에 따르면 고유한 키를 찾기 위해서는 r (= $\lceil$ key size / block size $\rceil$)개의 평문-암호문 쌍을 필요로 한다. 이러한 이유에서 블록 암호의 Grover 키 검색에 필요한 양자 자원을 계산할 때, Toffoli 게이트를 분해한 Clifford+T 수준의 양자 자원에 r을 곱한다. 키 크기와 블록 크기가 동일하거나 블록 크기가 키 크기보다 더 큰 경우 r은 1이 되기 때문에 무시할 수 있다. ARIA-128이 여기에 해당한다. 그러나 ARIA-192, ARIA-256의 경우 키 크기가 각각 192-bit, 256-bit로 블록 크기(128-bit)보다 크기 때문에 r이 2가 되어 r을 곱하는 것을 무시할 수 없다. 추가로 오라클 연산은 암호화 연산과 리버스 연산을 순차적으로 수행하기 때문에 오라클 회로 연산 1번을 수행할 때, ARIA의 암호 회로가 두 번 사용된다. 양자 회로의 자원값에 2를 곱하는 이유가 이에 해당한다. 마지막으로 Grover 검색 알고리즘은 $\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ 번 반복해줘야 하기 때문에 반복횟수만큼 또 양자 자원에 곱해야 한다. 결론적으로 ARIA에 대한 Grover 키 검색 비용은 [표 4-2]에 $r \times 2 \times \left\lfloor \frac{\pi}{4}\sqrt{2^k} \right\rfloor$ 를 곱한 것으로, [표 4-3]이 이에 대한 결과이다.

[표 4-3] ARIA 양자 회로에 대한 Grover 키 검색 비용 추정 결과 비교

Cipher	Source	Total gates (G)	Full-depth (FD)	Cost (complexity)	#qubit (M)
ARIA-128	CS	$1.946 \cdot 2^{85}$	$1.816 \cdot 2^{79}$	$1.767 \cdot 2^{165}$	1,561
	This work	$1.985 \cdot 2^{83}$	$1.626 \cdot 2^{76}$	$1.614 \cdot 2^{160}$	29,217
ARIA-192	CS	$1.133 \cdot 2^{119}$	$1.073 \cdot 2^{113}$	$1.216 \cdot 2^{232}$	3,121
	This work	$1.135 \cdot 2^{117}$	$1.949 \cdot 2^{109}$	$1.106 \cdot 2^{227}$	65,857
ARIA-256	CS	$1.371 \cdot 2^{151}$	$1.238 \cdot 2^{145}$	$1.698 \cdot 2^{296}$	3,377
	This work	$1.268 \cdot 2^{149}$	$1.092 \cdot 2^{142}$	$1.385 \cdot 2^{291}$	73,281

[표 4-3]의 비용(Cost)은 Grover 키 검색 비용을 의미하며, NIST에서 제공하는 보안 기준과 비교할 수 있는 지표이다. Grover 키 검색 비용과 2.4절에 기술된 양자 공격 비용($2^{157}, 2^{221}, 2^{285}$)을 비교해보면, 모든 ARIA 인스턴스(ARIA-128, ARIA-192, ARIA-256)가 각각의 키 크기에 적합한 수준의 보안을 달성하는 것을 확인할 수 있다.

본 논문에서는 Grover 키 검색 비용뿐 아니라, $TD-M$과 같은 지표를 포함한 평가를 수행하며, 이를 통해 Toffoli-depth와 큐비트 간의 trade-off를 평가한다. [표 4-3]에서 $TD-M$에 대해 Chauhan 과 본 논문에서 제안한 양자 회로의 결과를 비교해보면 이전보다 72.9%가 감소한 것을 알 수 있다. 이는 보조 큐비트의 할당을 통한 큐비트 수의 증가를 고려하더라도 Toffoli-depth가 충분히 감소하였음을 의미한다 볼 수 있다.

2.4절에서 언급한 NIST의 $MAXDEPTH$는 [표 4-3]의 Full-depth와 비교할 수 있다. [표 4-3]에 제시된 Full-depth(FD)와 NIST의 공식 문서에 제시된 $MAXDEPTH$를 비교해보면 ARIA-128만이 2^{76}로 $MAXDEPTH$ 요구사항을 충족한다 (ARIA-128$< 2^{96}$). ARIA-192와 ARIA-256처럼 Full-depth가 $MAXDEPTH$보다 큰 경우, $MAXDEPTH$를 고려하기 위해서는 병렬화를 간과할 수 없다. $MAXDEPTH$의 요건을 충족하지 못한 경우, Full-depth(FD)를 $(FD)/MAXDEPTH$만큼 줄이기 위해서 Grover 인스턴스는 $(FD)^2/(MAXDEPTH)^2$ 만큼 병렬로 작동해야 한다. 이 경우 $MAXDEPTH$는 원하는 만큼 감소시킬 수 있지만, 큐비트 수(M)가 $(FD)^2/(MAXDEPTH)^2$ 배 증가하기 때문에 최종적으로 Grover 키 검색 비용은 $(FD)^2/(MAXDEPTH)^2$ 가 아니라 $(FD)^2/(MAXDEPTH)^2 \times M$가 된다. 궁극적으로 Grover 검색의 병렬화를 고려한 $FD-M$ $(FD \times M)$ 비용은 $(FD)^2 \times M$이라 볼 수 있다. $(FD)^2 \times M$와 마찬가지로 $(TD)^2 \times M$도 Grover 검색의 병렬화를 고려한 $TD-M$ 비용을 의미한다. 추가적으로 $(FD)^2 \times G$ 는 Grover 검색 병렬화를 고려한 $FD-G$ 비용으로 FD^2 에 양자 게이트의 총 개수(G)를 곱한 것이다. $MAXDEPTH$ 제한을 고려하기 위해서는을 $(FD)^2 \times M$, $(FD)^2 \times G$, $(TD)^2 \times M$ 등 관련된 지표의 비용을 최소화

하는 방향을 생각해보아야 한다.

[표 4-4] *MAXDEPTH*를 고려한 ARIA 양자 회로에 대한 Grover 키 검색 비용 추정 결과

Cipher	Source	$TD-M$	$FD-M$	FD^2-M	FD^2-G
ARIA-128	CS	$1.26 \cdot 2^{87}$	$1.384 \cdot 2^{90}$	$1.257 \cdot 2^{170}$	$1.604 \cdot 2^{245}$
	This work	$1.313 \cdot 2^{84}$	$1.45 \cdot 2^{91}$	$1.179 \cdot 2^{168}$	$1.312 \cdot 2^{237}$
ARIA-192	CS	$1.489 \cdot 2^{121}$	$1.635 \cdot 2^{124}$	$1.754 \cdot 2^{237}$	$1.304 \cdot 2^{345}$
	This work	$1.672 \cdot 2^{119}$	$1.959 \cdot 2^{125}$	$1.908 \cdot 2^{236}$	$1.078 \cdot 2^{337}$
ARIA-256	CS	$1.921 \cdot 2^{153}$	$1.021 \cdot 2^{157}$	$1.264 \cdot 2^{302}$	$1.247 \cdot 2^{441}$
	This work	$1.04 \cdot 2^{152}$	$1.221 \cdot 2^{158}$	$1.409 \cdot 2^{300}$	$1.511 \cdot 2^{433}$

[표 4-4]에서 *MAXDEPTH*를 고려하지 않고 보면 Full depth의 증가폭에 비해 큐비트의 증가폭이 더 크게 느껴지기 때문에 최적화가 이뤄지지 않았다고 보여질 수도 있다. 그러나 *MAXDEPTH*를 고려한 지표인 $(FD)^2 \times M$를 보면 확실히 이전 결과보다 최적화가 진행된 것을 알 수 있다. 즉, *MAXDEPTH*를 고려한 상황에서 큐비트와의 적절한 trade-off를 가지는 depth 최적화된 회로를 구성했다고 볼 수 있는 것이다.

Ⅴ. 결론

본 논문에서는 회로의 depth 최적화에 중점을 두고 ARIA에 대한 최적화된 양자 회로를 제안하였다. 이진 필드에서 Toffoli-depth를 최적화한 Karatsuba 양자 곱셈, 선형 행렬에서의 최적화 기법, 병렬화와 같은 다양한 기술을 사용하여 합리적인 수의 큐비트를 보장하면서도 Toffoli-depth와 Full-depth를 감소시켰다. 결과적으로, 제안한 ARIA 양자 회로 구현은 선행 연구에 해당하는 SPACE'20에 게재된 Chauhan et al.의 논문에 제안된 구현 결과와 비교하여 개선율이 Full-depth의 경우 88.8% 이상, Toffoli-depth의 경우 98.7% 이상을 달성했다. 제안한 양자 회로를 기반으로 양자 자원과 Grover 키 검색 비용을 추정하였다. 그 후, Grover 키 검색 비용을 이용하여 NIST의 공식 문서에서 제공한 양자 보안 기준을 기반으로 양자 보안 강도를 평가했다. 그 결과, ARIA가 모든 인스턴스의 키 크기(128, 192, 256-bit)에 대해 양자 내성 보안 레벨 1, 3, 5를 각각 달성함을 확인하였다. 또한, *MAXDEPTH*도 고려한 결과, ARIA-128만이 *MAXDEPTH* 제한을 충족하였다. *MAXDEPTH* 충족하지 못하는 경우, *MAXDEPTH*를 고려하기 위한 관련 지표들에 대해서 언급하였고 해당 지표들을 비교한 결과 모두 이전의 결과보다 훨씬 최적화가 진행된 것을 확인하였다.

양자 컴퓨터의 개발이 진행됨에 따라 단순 Grover 키 검색 비용뿐 아니라 실제 실행과 관련이 깊은 *MAXDEPTH*의 중요성이 커질 것으로 사료된다. 따라서 4.2절의 마지막 부분에 언급한 *MAXDEPTH* 관련 내용과 연결 지어 양자 회로에 대해 *MAXDEPTH*의 제한을 고려한 최적화 방법을 모색해야 할 것이다.

참 고 문 헌

1. 국내문헌

KISA. (2004).『민관겸용 블록 암호 알고리즘 ARIA 알고리즘 명세서』. https://seed.kisa.or.kr/kisa/Board/19/detailView.do

2. 국외문헌

Kwon, D., Kim, J., Park, S., Sung, S. H., Sohn, Y., Song, J. H., ... & Hong, J. (2003, November). New block cipher: ARIA. In International conference on information security and cryptology (pp. 432-445). Berlin, Heidelberg: Springer Berlin Heidelberg.

Grover, L. K. (1996, July). A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (pp. 212-219).

Amy, M., Maslov, D., Mosca, M., & Roetteler, M. (2013). A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 32(6), 818-830.

Chauhan, A. K., & Sanadhya, S. K. (2020). Quantum resource estimates of grover's key search on aria. In Security, Privacy, and Applied Cryptography Engineering: 10th International Conference, SPACE 2020, Kolkata, India, December 17–21, 2020, Proceedings 10 (pp. 238-258). Springer International Publishing.

Jang, K., Kim, W., Lim, S., Kang, Y., Yang, Y., & Seo, H. (2023). Quantum Binary Field Multiplication with Optimized Toffoli Depth and Extension to Quantum Inversion. Sensors, 23(6), 3156.

Cheung, D., Maslov, D., Mathew, J., & Pradhan, D. K. (2008). On the design and optimization of a quantum polynomial-time attack on elliptic curve cryptography. In Theory of Quantum Computation, Communication, and Cryptography: Third Workshop, TQC 2008 Tokyo, Japan, January 30-February 1, 2008. Revised Selected Papers 3 (pp. 96-104). Springer Berlin Heidelberg.

Xiang, Z., Zeng, X., Lin, D., Bao, Z., & Zhang, S. (2020). Optimizing implementations of linear layers. IACR Transactions on Symmetric Cryptology, 120-145.

Chun, M., Baksi, A., & Chattopadhyay, A. (2023). Dorcis: Depth optimized quantum implementation of substitution boxes. Cryptology ePrint

Archive.

Chung, D., Lee, S., Choi, D., & Lee, J. (2021). Alternative tower field construction for quantum implementation of the aes s-box. IEEE Transactions on Computers, 71(10), 2553-2564.

Dasu, V. A., Baksi, A., Sarkar, S., & Chattopadhyay, A. (2019, September). LIGHTER-R: optimized reversible circuit implementation for sboxes. In 2019 32nd IEEE International System-on-Chip Conference (SOCC) (pp. 260-265). IEEE.

Fedorov, A., Steffen, L., Baur, M., da Silva, M. P., & Wallraff, A. (2012). Implementation of a Toffoli gate with superconducting circuits. Nature, 481(7380), 170-172.

Grassl, M., Langenberg, B., Roetteler, M., & Steinwandt, R. (2016, February). Applying Grover's algorithm to AES: quantum resource estimates. In International Workshop on Post-Quantum Cryptography (pp. 29-43). Cham: Springer International Publishing.

Itoh, T., & Tsujii, S. (1988). A fast algorithm for computing multiplicative inverses in GF (2m) using normal bases. Information and computation, 78(3), 171-177.

Jang, K., Baksi, A., Kim, H., Seo, H., & Chattopadhyay, A. (2022). Improved quantum analysis of speck and LOWMC (full version). Cryptology ePrint Archive.

Jaques, S., Naehrig, M., Roetteler, M., & Virdia, F. (2020). Implementing Grover oracles for quantum key search on AES and LowMC. In Advances in Cryptology-EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II 30 (pp. 280-310). Springer International Publishing.

Jang, K., Baksi, A., Kim, H., Song, G., Seo, H., & Chattopadhyay, A. (2022). Quantum analysis of aes. Cryptology ePrint Archive.

Jang, K., Kim, D., Oh, Y., Lim, S., Yang, Y., Kim, H., & Seo, H. (2023). Quantum Implementation of AIM: Aiming for Low-Depth. Cryptology ePrint Archive.

NIST.: Call for additional digital signature schemes for the post-quantum cryptography standardization process (2022) https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf.

Roy, S., Baksi, A., & Chattopadhyay, A. (2023). Quantum implementation of ascon linear layer. Cryptology ePrint Archive.

Yang, Y., Jang, K., Kim, H., Song, G., & Seo, H. (2022, August). Grover on SPARKLE. In International Conference on Information Security Applications (pp. 44-59). Cham: Springer Nature Switzerland.

ABSTRACT

Depth-Optimized Quantum Circuit Implementation for Block Cipher ARIA

Yang, Yu-Jin

Major in IT Convergence Engineering

Dept. of IT Convergence Engineering

The Graduate School

Hansung University

The advancement of large-scale quantum computers poses a threat to the security of current encryption systems. In particular, symmetric-key cryptography significantly is impacted by general attacks using the Grover's search algorithm. In recent years, studies have been presented to estimate the complexity of Grover's key search for symmetric-key ciphers and assess post-quantum security. In this paper, we propose a depth-optimized quantum circuit implementation for ARIA, which is a symmetric key cipher included as a validation target the Korean Cryptographic Module Validation Program (KCMVP). Our quantum circuit implementation for ARIA improves the depth by more than 88.8% and Toffoli-depth by more than 98.7% compared to the implementation presented in Chauhan et al.'s SPACE'20 paper. Finally, we present the

cost of Grover's key search for our circuit and evaluate the post-quantum security strength of ARIA according to relevant evaluation criteria provided NIST.

【Keywords】ARIA Block Cipher, Depth-Optimized Quantum Circuit Implementation, Grover's Search Algorithm, Evaluation of Post-quantum Security Strength