논문 리뷰

Quantum Resource Estimates of Grover's Key Search on ARIA

발표자: 양유진

링크: https://youtu.be/vUaKgx1ohYc

1. Quantum Circuits to Implement ARIA - Substitution layer

$$S_1(\alpha) := \mathbf{A}.\alpha^{-1} + \mathbf{a}$$

$$\mathbf{A} = \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

S_1 은 AES의 S-box와 같음.

$$S_2(\alpha) := \mathbf{B}.\alpha^{247} + \mathbf{b}$$

$$:= \mathbf{B}.(\alpha^{-1})^8 + \mathbf{b} = \mathbf{B}.\mathbf{C}.\alpha^{-1} + \mathbf{b}$$

$$= \mathbf{D}.\alpha^{-1} + \mathbf{b}$$

$$\mathbf{B} = \begin{bmatrix} 0&1&0&1&1&1&1&0 \\ 0&0&1&0&1&1&1&1 \\ 1&0&0&1&0&1&1&1 \\ 1&1&0&0&1&0&1&1 \\ 1&1&1&1&0&0&1&0 \\ 0&1&1&1&1&0&0&1 \\ 1&0&1&1&1&1&0&0 \\ 0&1&0&1&1&1&1&0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0&1&0&1&0&1&1&1 \\ 0&0&1&1&1&1&1&1 \\ 1&1&1&0&1&1&0&1 \\ 1&1&0&0&0&0&1&1 \\ 0&1&0&0&0&0&1&1 \\ 1&1&0&0&1&1&1&0 \\ 0&1&1&0&0&0&1&1 \\ 1&1&1&1&0&1&1&0 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

1. Quantum Circuits to Implement ARIA - Substitution layer

1) α^{-1} 계산

- Itoh-Tsujii multiplier 사용하여 역원 구함

$$\alpha^{-1} = \alpha^{254} = ((\alpha.\alpha^2).(\alpha.\alpha^2)^4.(\alpha.\alpha^2)^{16}.\alpha^{64})^2$$

(1) 제곱기(squaring)

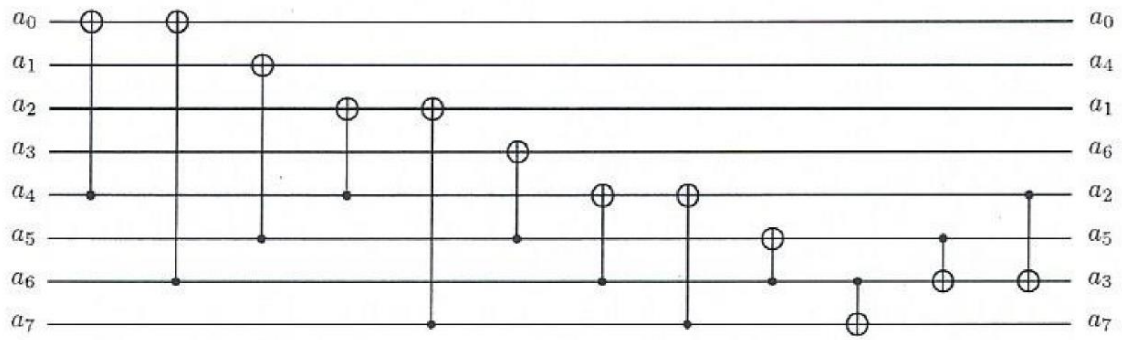


Fig. 1. Circuit for squaring in $\mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$.

- CNOT 게이트 12개 사용
- depth: 7

1. Quantum Circuits to Implement ARIA - Substitution layer

(2) 곱셈기(multiplier)

- schoolbook multiplier(Maslov et al.) 사용

$\boldsymbol{a} = [a_0, \cdots . a_7]^{\mathrm{T}}$ $\mathbf{b} = [b_0, \cdots . b_7]^{\mathrm{T}}$ $\mathbf{c} = \mathbf{a} \cdot \mathbf{b}$

$\mathbf{c} = [c_0, \cdots . c_7]^{\mathrm{T}}$

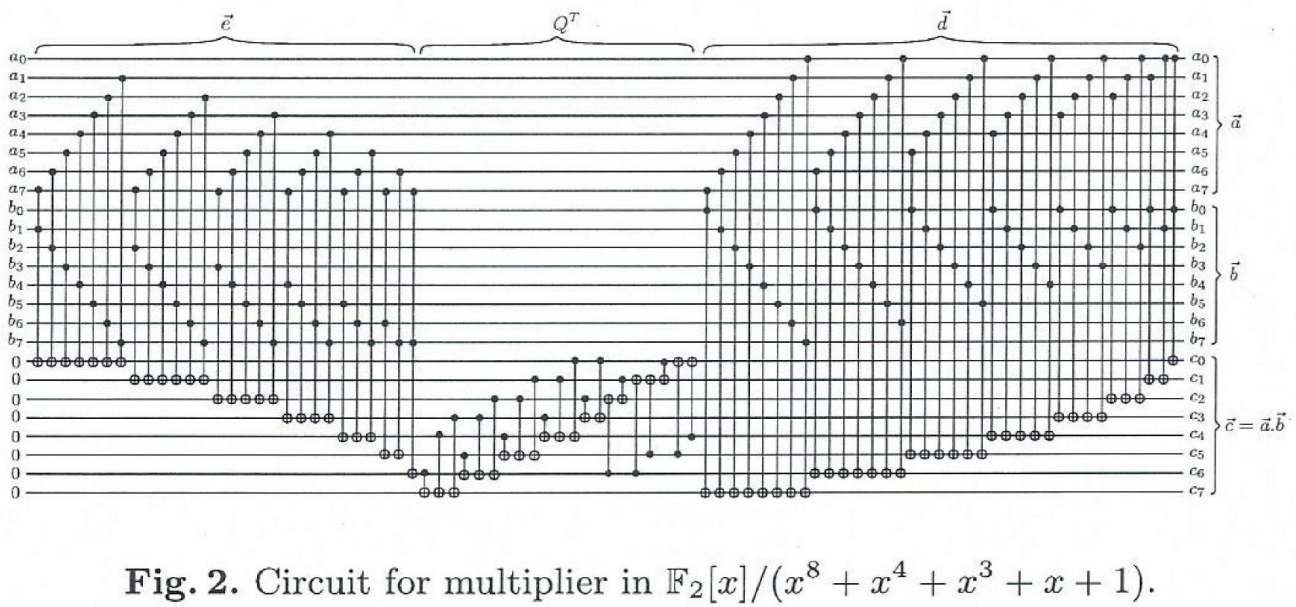


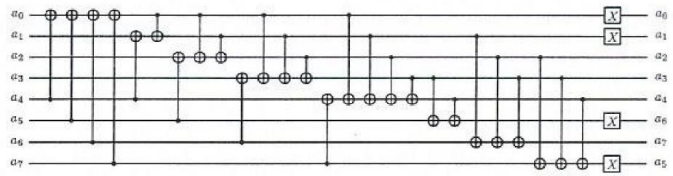
Fig. 2. Circuit for multiplier in $\mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$.

- CNOT 게이트 21개 사용
- Toffoli 게이트 64개 사용
- depth: 37
- Toffoli-depth: 28

1. Quantum Circuits to Implement ARIA - Substitution layer

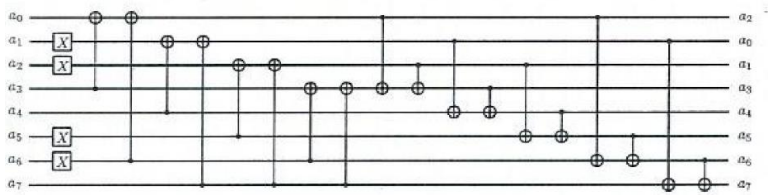
2) Affine function

- S-box에 따라 다른 회로가 사용됨.



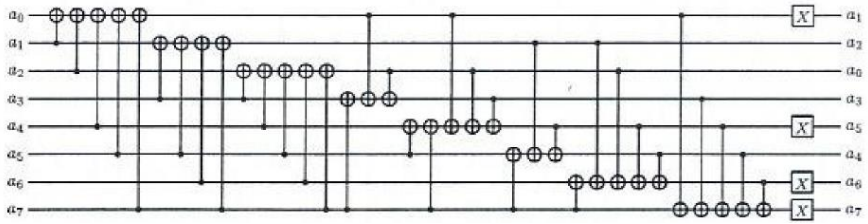
S_1 - CNOT 게이트 26개 사용

- Pauli-X 게이트 4개 사용



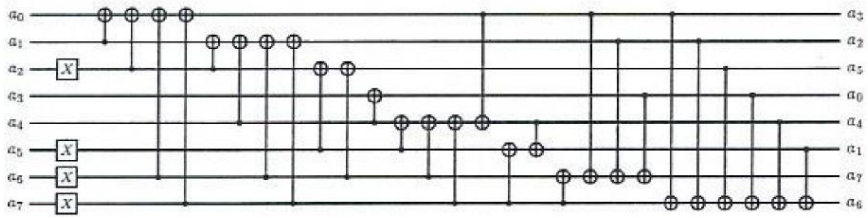
S_1^{-1} - CNOT 게이트 18개 사용

- Pauli-X 게이트 4개 사용



S_2 - CNOT 게이트 35개 사용

- Pauli-X 게이트 4개 사용



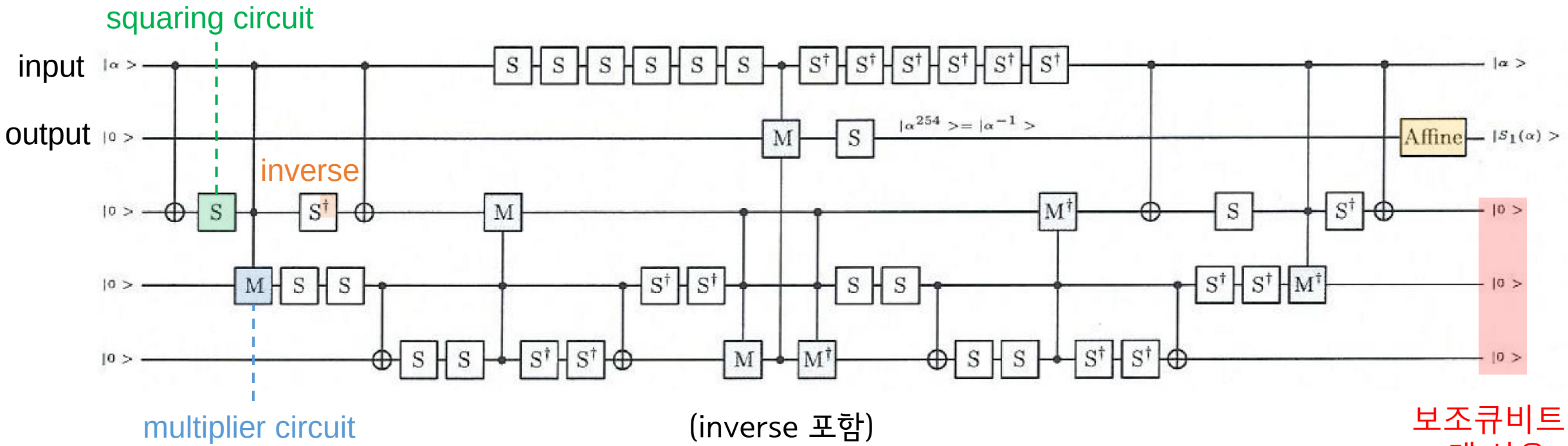
S_2^{-1} - CNOT 게이트 27개 사용

- Pauli-X 게이트 4개 사용

1. Quantum Circuits to Implement ARIA - Substitution layer

3) SubBytes function

- SubBytes 함수는 S_1와 S_2 가 같음



(inverse 포함)

보조큐비트
3개 사용

- squaring 33번 사용
- multiplier 7번 사용
- 큐비트 40개 필요

1. Quantum Circuits to Implement ARIA - Substitution layer

S_1 양자 게이트 수

- Toffoli 게이트: 64 (곱셈기 1번) x 7 (S_1에 7번 사용) = 448
- CNOT 게이트: 12 x 33 (제곱기) + 21 x 7 (곱셈기) + 26 (아핀 변환) = 569
- Pauli-X 게이트: 4 (아핀 변환)

S_1^{-1} 양자 게이트 수

- Toffoli 게이트: 448
- CNOT 게이트: 543 + 18 (아핀 변환) = 561
- Pauli-X 게이트: 4 (아핀 변환)

S_2 양자 게이트 수

- Toffoli 게이트: 448
- CNOT 게이트: 543 + 35 (아핀 변환) = 578
- Pauli-X 게이트: 4 (아핀 변환)

S_2^{-1} 양자 게이트 수

- Toffoli 게이트: 448
- CNOT 게이트: 543 + 27 (아핀 변환) = 570
- Pauli-X 게이트: 4 (아핀 변환)

substitution layer 양자 게이트 수

- Toffoli 게이트: 448 x (4 x 4) = 7,618
- CNOT 게이트: (569 + 561 + 578 + 570) x 4 = 9,112
- Pauli-X 게이트: 4 x (4 x 4) = 64

1. Quantum Circuits to Implement ARIA - Diffusion layer

- 16 x 16 matrix → 128 x 128 matrix 변환

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{bmatrix} = \begin{bmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix}$$

0 이면

$$\begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{bmatrix}$$

1 이면

$$\begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

- PLU decomposition을 사용함.
  - CNOT 게이트 768개 사용
  - depth: 26

1. Quantum Circuits to Implement ARIA - Add Round Key

- 128-bit의 라운드키를 현재 상태와 XOR 함.
- XOR 구현에 CNOT 게이트가 사용됨.
- 1번 연산할 때 128개의 CNOT 게이트가 사용됨.
- 128개의 CNOT 게이트를 병렬로 실행할 수 있기 때문에 회로의 깊이는 1.
- 첫번째 라운드 전에는 초기 ARK 연산이 적용됨
- 마지막 라운드에서는 DL(Diffusion Layer) 대신 ARK 연산이 적용됨.

1라운드 양자 게이트 수

- Toffoli 게이트: 7,618
- CNOT 게이트: 128(ARK) + 9,112(SL) + 768(DL) = 10,008
- Pauli-X 게이트: 64

1. Quantum Circuits to Implement ARIA - Key Schedule

- W_1, W_2, W_3 의 비용은 3 라운드 Feistel 암호(F_o, F_e) 연산과 관련이 있음.
- 각 라운드 키 계산과 큐비트 절약을 위한 상태의 uncompute 연산에 1개의 큐비트 상태 W_4만 사용함.
  → 각 라운드 키 생성에 512 개의 CNOT게이트가 사용됨.
- rotation operation은 비용이 발생하지 않음

Table 2. Quantum cost of generating four quantum words and round subkeys for the key schedule of ARIA-{128, 192, 256}.

KeyWords (W_i)	# Pauli-X	# CNOT	# Toffoli
W_0	0	128	0
W_1	$64 + 65 = 129$	$10,008 + 128 = 10,136$	7,168
W_2	$64 + 65 = 129$	$10,008 + 128 = 10,136$	7,168
W_3	$64 + 57 = 121$	$10,008 + 128 = 10,136$	7,168
Total	379	30,536	21,504
Round Subkeys	# Pauli-X	# CNOT	# Toffoli
RK_i for each i	0	$128 \times 4 = 512$	0

초기화 상수 CK_i 생성에 사용됨.

2. Resource Estimates: Reversible ARIA Implementation

- “zig-zag” 방식을 사용하여 qubit 최적화에 집중함
- ARIA-128 기준으로 키 생성, 암호화에 큐비트가 1,408 개 사용되었고,
  보조큐비트(SubBytes에 사용)는 24개 사용됨.
- 보조큐비트는 라운드 키 생성과 암호화 라운드 사이에 위치함.
- 암호화 단계의 경우 5, 9, 12 라운드 후에 reverse 연산이 수행되어야 하기 때문에
  640-bit(ARIA-128/192), 768-bit(ARIA-256) 의 저장공간이 필요함.

2. Resource Estimates: Reversible ARIA Implementation

Table 3. Quantum resource estimates for the implementation of ARIA-128.

Phase	#Quantum Gates			Depth		#Qubits	
	#Pauli-X	#CNOT	#Toffoli	Toffoli	Overall	Storage	Ancilla
Initial	0	0	0	0	0	256	0
Key Gen	379	41,228	21,504	588	1,342	512	128
Encryption	1,216	189,896	136,192	3,724	7,918	640	24
Total	1,595	231,124	157,696	4,312	9,260	1,408	152

Table 4. Quantum resource estimates for the implementation of ARIA-192.

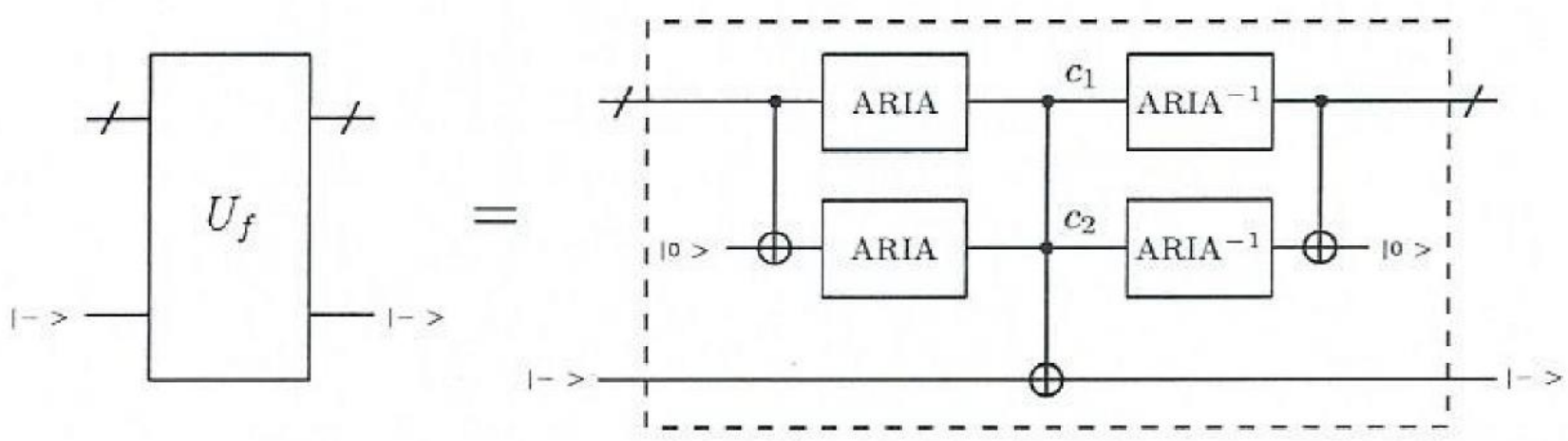
Phase	#Quantum Gates			Depth		#Qubits	
	#Pauli-X	#CNOT	#Toffoli	Toffoli	Overall	Storage	Ancilla
Initial	0	0	0	0	0	256	0
Key Gen	379	43,336	21,504	588	1,358	512	128
Encryption	1,472	229,928	164,864	4,508	9,590	640	24
Total	1,851	273,264	183,368	5,096	10,948	1,408	152

Table 5. Quantum resource estimates for the implementation of ARIA-256.

Phase	#Quantum Gates			Depth		#Qubits	
	#Pauli-X	#CNOT	#Toffoli	Toffoli	Overall	Storage	Ancilla
Initial	0	0	0	0	0	256	0
Key Gen	379	45,384	21,504	588	1,374	512	128
Encryption	1,792	279,968	200,704	5,488	11,680	768	24
Total	2,171	325,352	222,208	6,076	13,054	1,536	152

3. Grover Oracle and Key Search Resource Estimates

- Langenberg et al. 에 의해 $\boldsymbol{r_k = [k/n]}$
  (r_k는 필요로 하는 쌍 개수, k는 키 길이, n은 평문 길이)



[ARIA-128에 대한 함수 U_f의 가역 구현 (r=2)]

- ARIA-128: $r_{128} = 1$ 평문-암호문 쌍 → ARIA 인스턴스 2개
- ARIA-192: $r_{128} = 2$ 평문-암호문 쌍 → ARIA 인스턴스 4개
- ARIA-256: $r_{256} = 2$ 평문-암호문 쌍 → ARIA 인스턴스 4개

3. Grover Oracle and Key Search Resource Estimates

[표] Grover Oracle에 필요한 양자 자원 추정치

Parameter	Toffoli gates	Toffoli depth	CNOT	Pauli-X
ARIA - 128	315,392	8,624	462,248	3,190
ARIA - 192	733,472	20,384	1,093,056	7,404
ARIA - 256	888,832	24,304	1,301,408	8,684

$$G = U_f\left(\left(H^{\otimes k}\left(2\left|0\right\rangle\left\langle 0\right| - \mathbf{1}_{2^k}\right)H^{\otimes k}\right) \otimes \mathbf{1}_2\right)$$

$\left\lfloor \frac{\pi}{4} \cdot \sqrt{2^k} \right\rfloor$번 반복

- **Toffoli-gate = $\mathbf{7} \times \boldsymbol{T} + \mathbf{8} \times \boldsymbol{Clifford}$** (T-depth=4, total depth=8)
- l-fold를 구현하기 위한 T-gate의 CNOT 게이트는 $\mathbf{(32} \times \boldsymbol{l} - \mathbf{84)}$ 로 추정 ($l \geq 5$)
- **Clifford-gate = $\mathbf{2} \cdot (\boldsymbol{r_k} - \mathbf{1}) \cdot \boldsymbol{k}$ CNOT** (병렬 처리를 위해 U_f 내부의 CNOT 게이트만 계산)

4. Cost Comparison of ARIA and AES

- G-cost: 전체 게이트 수
- DW-cost: circuit depth x width

Jaques et al.'s work [13]									
Scheme	r_k	#Clifford	$\#T$	T-depth	full depth width		G-cost	DW-cost	p_s
AES-128	1	$1.03 \cdot 2^{85}$	$1.59 \cdot 2^{84}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	984	$1.83 \cdot 2^{85}$	$1.11 \cdot 2^{91}$	$1/e$
AES-192	2	$1.17 \cdot 2^{118}$	$1.81 \cdot 2^{117}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	2224	$1.04 \cdot 2^{119}$	$1.44 \cdot 2^{124}$	1
AES-256	2	$1.46 \cdot 2^{150}$	$1.13 \cdot 2^{150}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	2672	$1.30 \cdot 2^{151}$	$1.02 \cdot 2^{157}$	$1/e$

This Work									
Scheme	r_k	#Clifford	$\#T$	T-depth	full depth width		G-cost	DW-cost	p_s
ARIA-128	1	$1.11 \cdot 2^{85}$	$1.65 \cdot 2^{84}$	$1.65 \cdot 2^{78}$	$1.81 \cdot 2^{79}$	1561	$1.93 \cdot 2^{85}$	$1.37 \cdot 2^{90}$	$1/e$
ARIA-192	2	$1.30 \cdot 2^{118}$	$1.92 \cdot 2^{117}$	$1.95 \cdot 2^{111}$	$1.07 \cdot 2^{112}$	3121	$1.13 \cdot 2^{119}$	$1.63 \cdot 2^{123}$	1
ARIA-256	2	$1.57 \cdot 2^{150}$	$1.16 \cdot 2^{150}$	$1.16 \cdot 2^{144}$	$1.23 \cdot 2^{144}$	3377	$1.36 \cdot 2^{151}$	$1.01 \cdot 2^{156}$	$1/e$

G-cost는 둘이 거의 비슷하지만, DW-cost는 ARIA가 AES보다 더 낮음

5. Conclusion

- ARIA를 직접 구현하여 각 단계에서 발생하는 양자 자원 비용을 추정함
- ARIA 3가지 인스턴스에 대한 Grover 검색 공격 비용 제공함

Future works

- Grover 검색 공격 복잡성을 줄이는 것
- 자동 리소스 추정을 위해 양자 프로그래밍 언어로 ARIA용 Grover oracle 구현하는 것
- 블록암호를 다중 대상 공격(multi-target attacks)에 대해 구현하여 비용 평가하는 것

감사합니다