ASCON 양자 구현 및 분석

ASCON 양자 회로 depth 최적 구현 및 분석

한성대학교 융합보안학과 23213702 오유진

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

서론

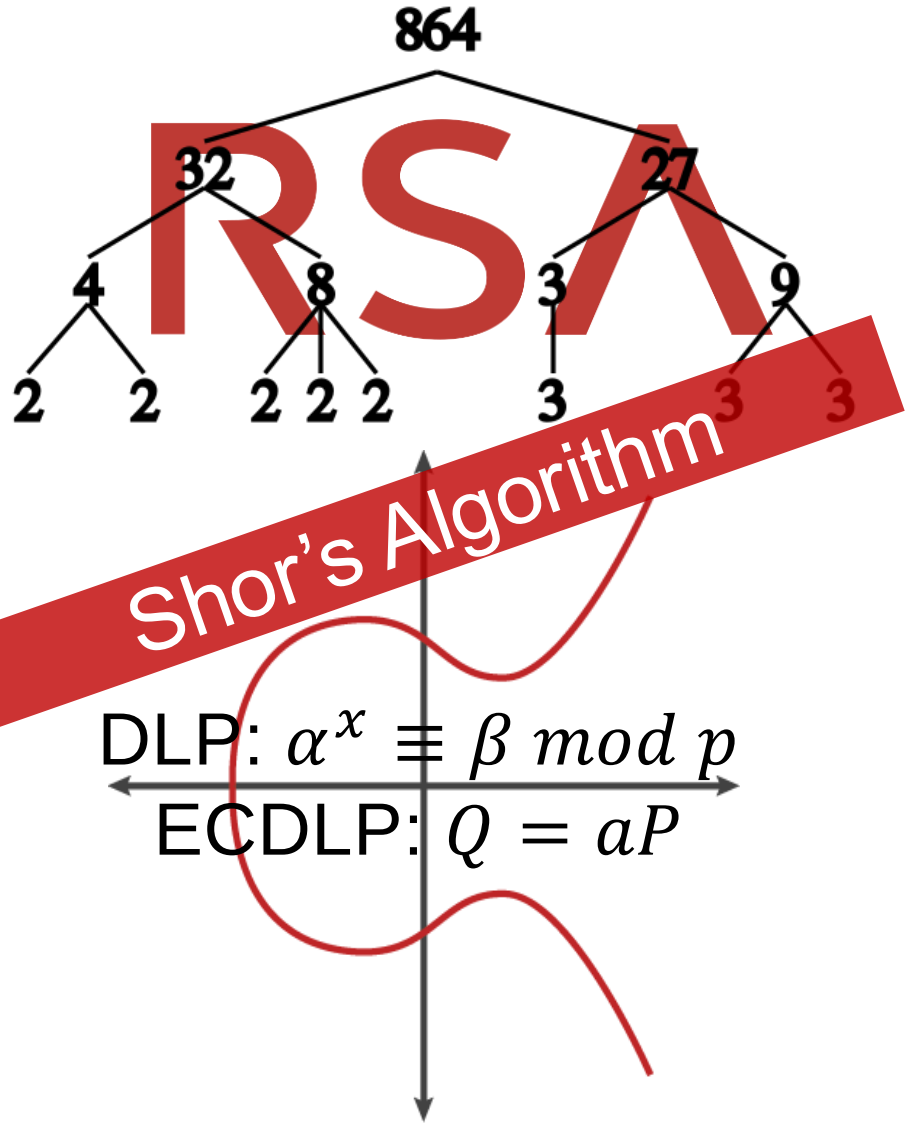
관련연구

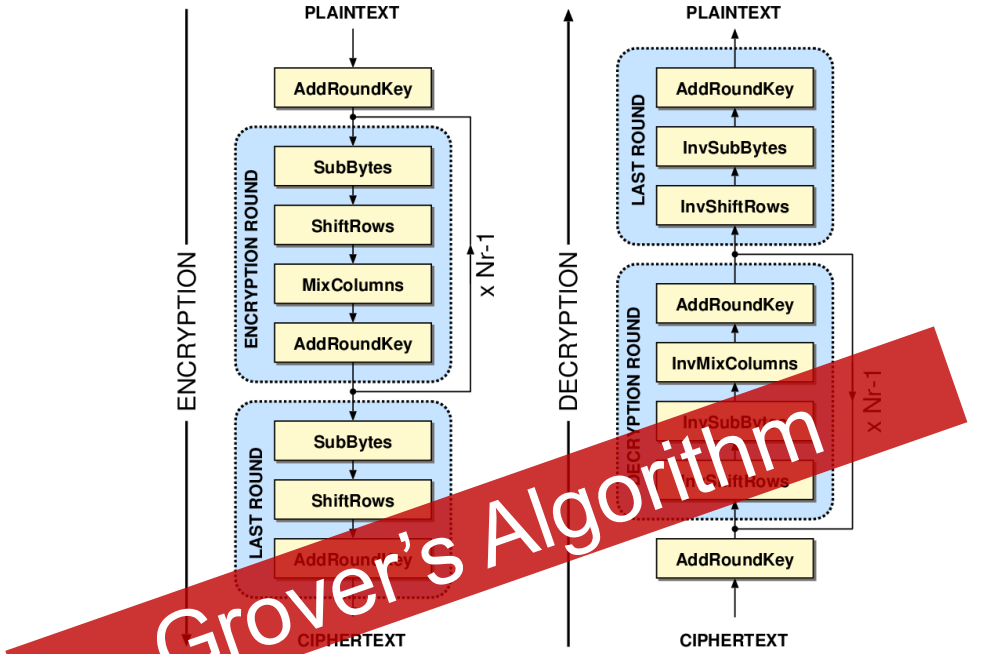
제안 기법

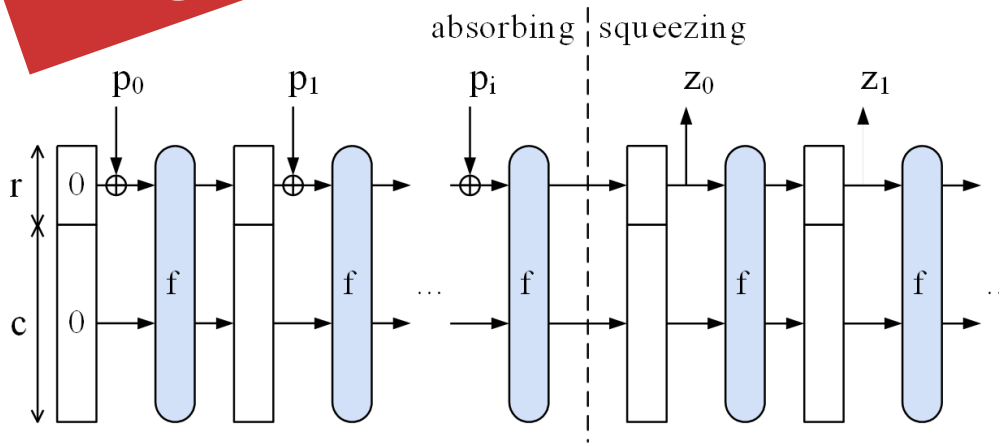
성능 평가

결론

서론


864
32
27
4
8
3
9
2
2
2
2
2
3
3
3
RSA
Shor's Algorithm
DLP: $\alpha^x \equiv \beta \ mod \ p$
ECDLP: $Q = aP$


PLAINTEXT
AddRoundKey
SubBytes
ShiftRows
MixColumns
AddRoundKey
ENCRYPTION ROUND
x Nr-1
SubBytes
ShiftRows
AddRoundKey
LAST ROUND
ENCRYPTION
CIPHERTEXT
PLAINTEXT
AddRoundKey
InvSubBytes
InvShiftRows
LAST ROUND
AddRoundKey
InvMixColumns
InvSubBytes
InvShiftRows
ECRYPTION ROUND
x Nr-1
AddRoundKey
DECRYPTION
CIPHERTEXT
Grover's Algorithm


absorbing
squeezing
p0
p1
pi
z0
z1
r
c
0
0
f
f
f
f
f
f

관련 연구

관련 연구 : ASCON

- **ASCON** - NIST 경량암호 표준화로 선정된 암호 제품군
  - 모든 체계는 128비트 보안을 제공하고 내부적으로 동일한 320비트 순열을 사용하므로 AEAD와 Hash모두 구현하기 좋음.

- ASCON has two variants
  - **ASCON-AEAD**
  - **a hash function**

- ASCON – AEAD
  - ASCON-128
  - ASCON-128a

- ASCON 해시 함수
  - ASCON-HASH
  - ASCON-XoF

Name	Algorithms	Bit size				Rounds	
		Key	Nonce	Tag	block	p^a	p^b
ASCON-128	$\varepsilon, D_{128,64,12,6}$	128	128	128	64	12	6
ASCON-128a	$\varepsilon, D_{128,128,12,8}$	128	128	128	128	12	8

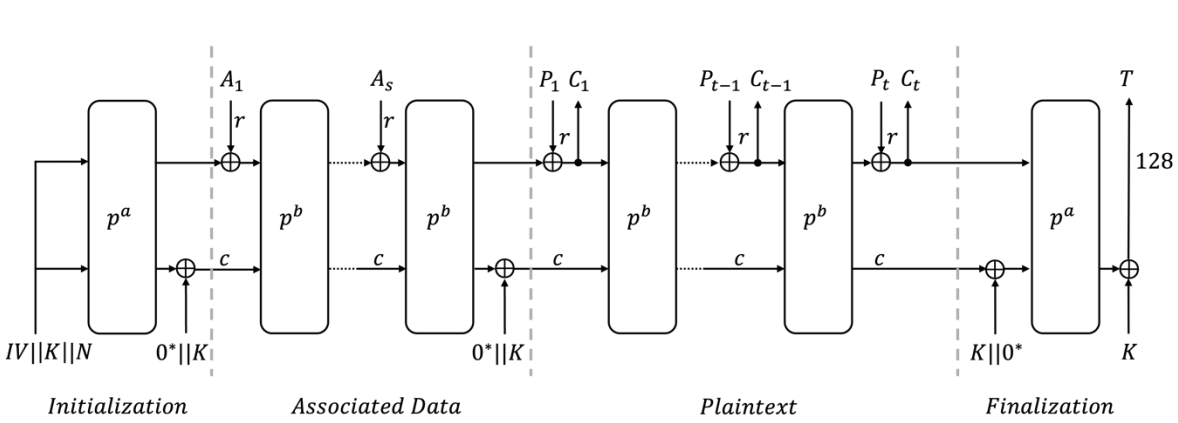
Name	Algorithms	Bit size		Rounds
		Hash	block	p^a
ASCON-Hash	$\chi_{256,64,12}$ with $l=256$	256	64	12
ASCON-Xof	$\chi_{0,64,12}$ with arbitrary l	l	64	12

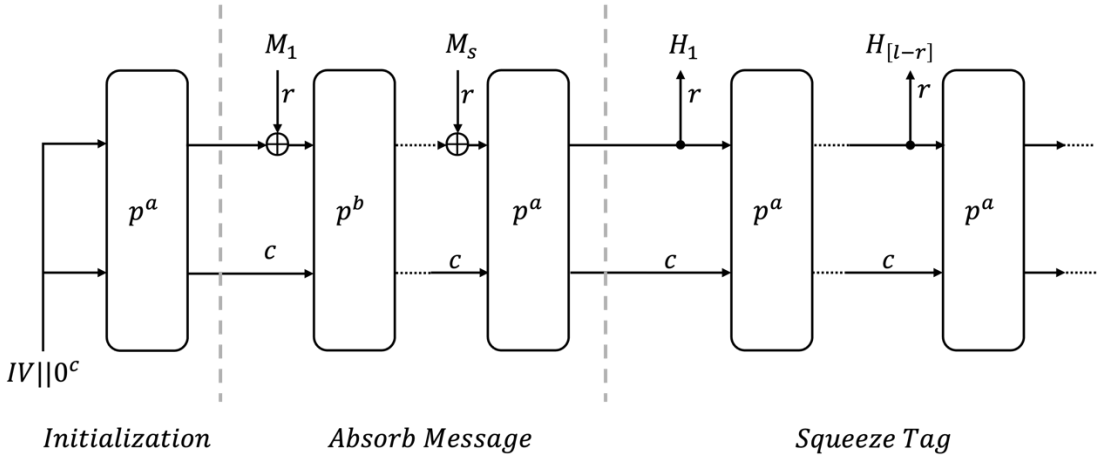
관련 연구 : ASCON

- ASCON-AEAD : *Initialization, processing Associated Data, Processing Plaintext, Finalization*
- ASCON-HASH : *Initialization, Absorbing, Squeezing*
- ASCON의 주요 구성요소는 320비트 단위로 동작하는 순열 함수

$P = P_L \circ P_S \circ P_C$

320비트 S는 5개의 64비트 레지스터 워드 $x_i, S = x_0 \parallel x_1 \parallel x_2 \parallel x_3 \parallel x_4$로 분할 ($x_0 = MSB$, $x_4 = LSB$)





관련 연구 : Grover 알고리즘

- **Grover 알고리즘**

1. 입력 설정

- 하다마드 게이트를 사용하여 k-큐비트 키를 중첩 상태로 만든다 (2^k 개의 모든 키가 동일한 진폭을 가짐)

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

2. 오라클 단계

- 이전 단계에서 준비된 중첩상태의 키를 사용하여 알려진 평문을 중첩 상태에서 암호화

  → 모든 가능한 키 값에 대한 암호문이 생성됨

  → 생성된 암호문은 알려진 암호문과 비교되며, 일치하는 경우($f(x) = 1$) 키 값의 부호 반전 ($(-1)^{f(x)}$)

- 구현된 양자 회로는 다음 반복을 위해 역연산되어 생성된 암호문을 알려진 평문으로 다시 변환

$$f(x) = \begin{cases} 1 \text{ if } Enc_{key}(p) = c \\ 0 \text{ if } Enc_{key}(p) \neq c \end{cases} \qquad U_f(|\psi\rangle|-\rangle) = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle|-\rangle$$

3. 확산 연산자는 오라클에서 반환된 솔루션 키의 진폭을 증폭 시킴

관련 연구 : Grover 양자 충돌 공격

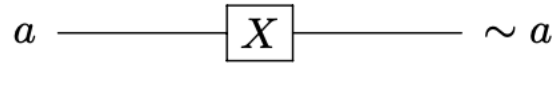
- **Grover 알고리즘을 이용한 양자 충돌 공격**
  - Grover 알고리즘을 이용한 다양한 양자 충돌 공격 존재

  - BHT 알고리즘
    - 공격 복잡도 $O\left(2^{\frac{n}{3}}\right)$, 양자 메모리 $O\left(2^{\frac{2n}{3}}\right)$

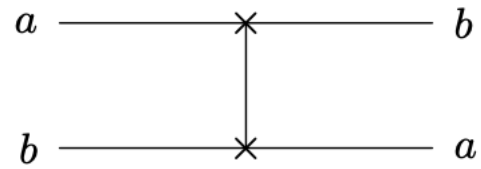
  - **CNS 알고리즘**
    - 공격 복잡도 $O\left(2^{\frac{2n}{5}}\right)$, 고전 메모리 $O\left(2^{\frac{n}{5}}\right)$
    - 병렬처리를 통해 공격 복잡도를 감소 시킬 수 있음
    - $2s$ 개의 양자 인스턴스를 병렬처리에서 활용
      - 공격 복잡도 $\boldsymbol{O\left(2^{\frac{2n}{5}-\frac{3s}{5}}\right)}$로 감소 $s \leq \frac{n}{4}$
      - Jang et al. 의 논문에서 $s = \frac{n}{6}$ 로 정의 → 본 논문에서도 해당 접근방식을 따름

관련 연구 : 양자 게이트

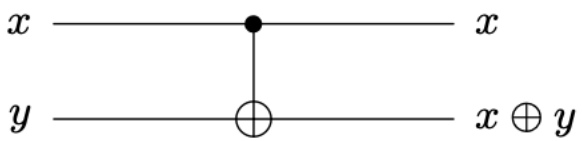
- **양자 게이트**



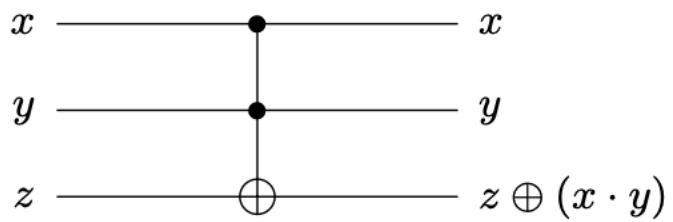
(a) X gate



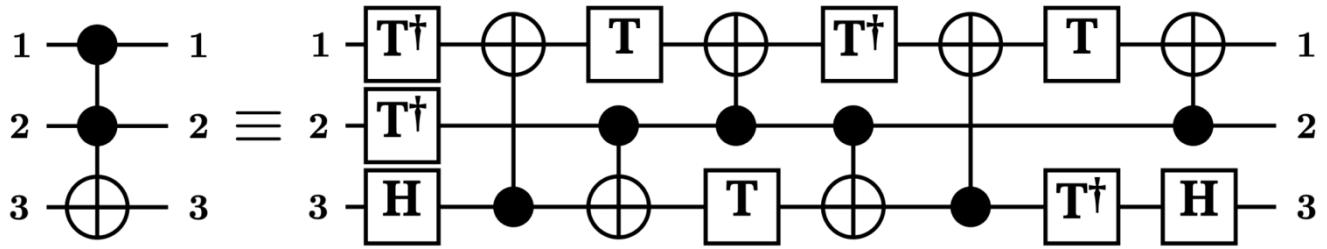
(b) Swap gate



(c) CNOT gate



(d) Toffoli gate



Toffoli gate decomposition (T- depth 4, total depth 8)

관련 연구 : NIST 보안레벨

- **NIST 양자 후 보안 레벨**
  - NIST는 AES 및 SHA-2/3 제품군에 대해 각각 Grover 키 검색과 충돌 검색 복잡도를 기반으로 **보안 레벨 정의**
  - Level 1, 3, 5는 AES에 대한 Grover 키 검색 복잡도에 해당
  - Level 2, 4는 SHA-2/3에 대한 충돌 검색 복잡도
    - 양자 공격 비용은 아직 정의되지 않았으며, 고전적인 공격 비용만 정의되어있음
    - **Jang et al. 이 정의한 보안레벨과 비교**

보안 레벨	암호	비용(복잡도)
Level 1	AES-128	$2^{170} \rightarrow 2^{157}$
Level 2	SHA-256/SHA3-256	2^{146} (classical gates)
Level 3	AES-192	$2^{233} \rightarrow 2^{221}$
Level 4	SHA-384/SHA3-384	2^{210} (classical gates)
Level 5	AES-256	$2^{198} \rightarrow 2^{285}$

[표 2-3] NIST 양자 후 보안 레벨

Level	Cipher	Cost
Level 2	SHA-2/3 (256)	$2^{188}/2^{183}$
Level 4	SHA-2/3 (384)	$2^{266}/2^{260}$
Level 6 (Extension)	SHA-2/3 (512)	$2^{343}/2^{337}$

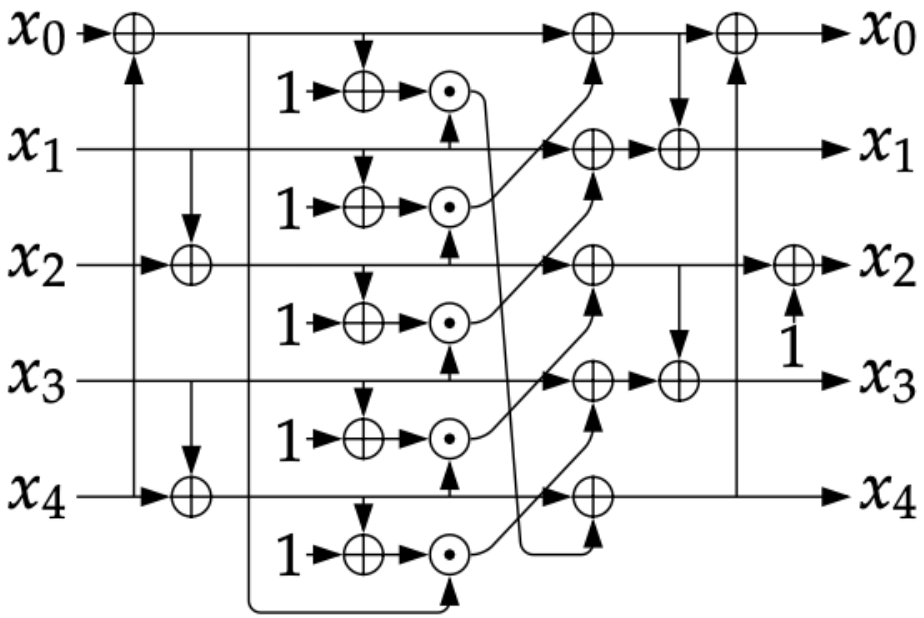
[표 2-4] Jang et al.이 정의한 양자 충돌 공격에 대한 보안 레벨

Jang, K., Lim, S., Oh, Y., Kim, H., Baksi, A., Chakraborty, S., & Seo, H. (2024). Quantum Implementation and Analysis of SHA-2 and SHA-3. Cryptology ePrint Archive.

제안 기법

제안 기법

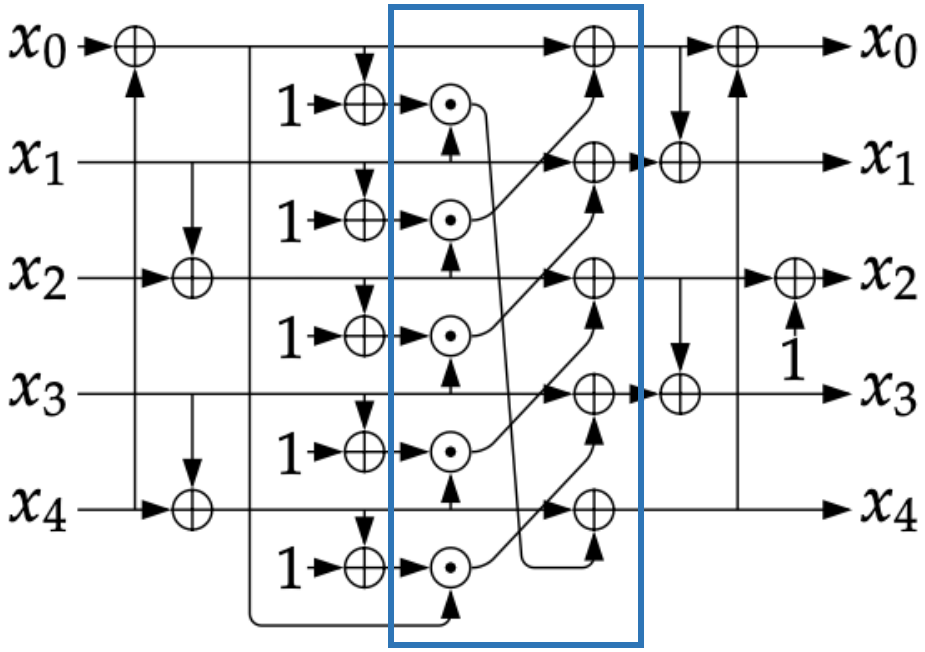
- **S-box 병렬 구현**
  - 양자 컴퓨터의 가역적인 특성으로 인해 look-up 테이블을 사용 할 수 없음
  - S-box 양자 회로는 **양자 게이트**를 사용하여 **부울 표현식**을 기반으로 구현해야 함.



$$
\begin{aligned}
&x_0 = x_0 \oplus x_4, \quad x_4 = x_4 \oplus x_3, \quad x_2 = x_2 \oplus x_1, \\
&t_0 = x_0, \quad t_1 = x_1, \quad t_2 = x_2, \quad t_3 = x_3, \quad t_4 = x_4, \\
&t_0 = \sim t_0, \quad t_1 = \sim t_1, \quad t_2 = \sim t_2, \quad t_3 = \sim t_3, \quad t_4 = \sim t_4, \\
&t_0 = t_0 \cdot x_1, \quad t_1 = t_1 \cdot x_2, \quad t_2 = t_2 \cdot x_3, \quad t_3 = t_3 \cdot x_4, \quad t_4 = t_4 \cdot x_0, \\
&x_0 = x_0 \oplus t_1, \quad x_1 = x_1 \oplus t_2, \quad x_2 = x_2 \oplus t_3, \quad x_3 = x_3 \oplus t_4, \quad x_4 = x_4 \oplus t_0, \\
&x_1 = x_1 \oplus x_0, \quad x_0 = x_0 \oplus x_4, \quad x_3 = x_3 \oplus x_2, \quad x_2 = \sim x_2.
\end{aligned} \tag{3}
$$

제안 기법

- **S-box 병렬 구현**
  - AND연산과 XOR 연산을 위해 $t_0 \sim t_4$를 위한 보조 큐비트가 필요
  - 64개의 S-box 사용→ 총 **320(5 x 64)**개의 보조 큐비트 필요
  - 이 경우, 연산이 **순차적**으로 이루어지므로 **Toffoli depth 증가**

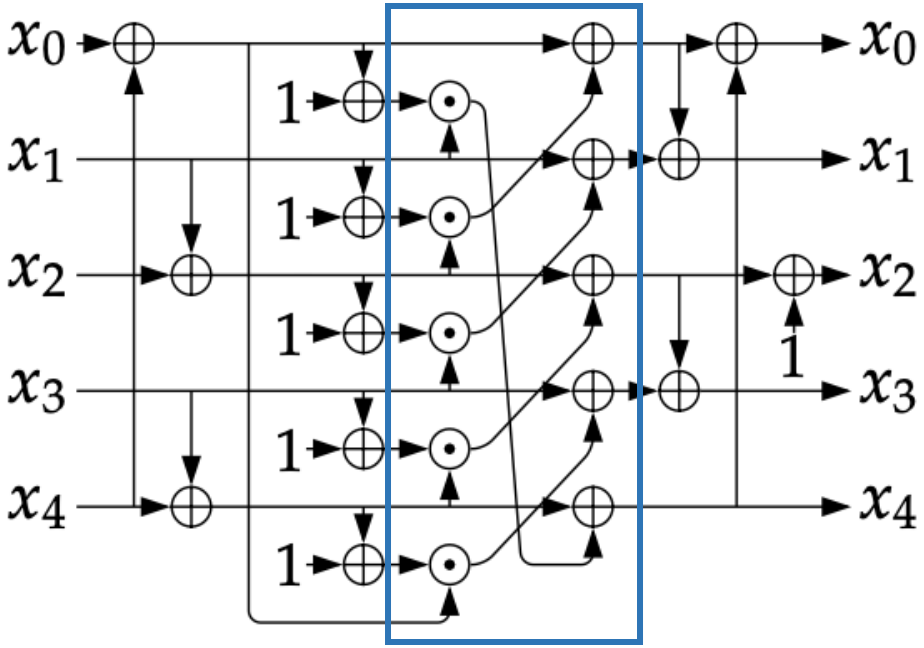


$$
\begin{aligned}
&x_0 = x_0 \oplus x_4, \quad x_4 = x_4 \oplus x_3, \quad x_2 = x_2 \oplus x_1, \\
&t_0 = x_0, \quad t_1 = x_1, \quad t_2 = x_2, \quad t_3 = x_3, \quad t_4 = x_4, \\
&t_0 = \sim t_0, \quad t_1 = \sim t_1, \quad t_2 = \sim t_2, \quad t_3 = \sim t_3, \quad t_4 = \sim t_4, \\
&t_0 = t_0 \cdot x_1, \quad t_1 = t_1 \cdot x_2, \quad t_2 = t_2 \cdot x_3, \quad t_3 = t_3 \cdot x_4, \quad t_4 = t_4 \cdot x_0, \\
&x_0 = x_0 \oplus t_1, \quad x_1 = x_1 \oplus t_2, \quad x_2 = x_2 \oplus t_3, \quad x_3 = x_3 \oplus t_4, \quad x_4 = x_4 \oplus t_0, \\
&x_1 = x_1 \oplus x_0, \quad x_0 = x_0 \oplus x_4, \quad x_3 = x_3 \oplus x_2, \quad x_2 = \sim x_2.
\end{aligned} \tag{3}
$$

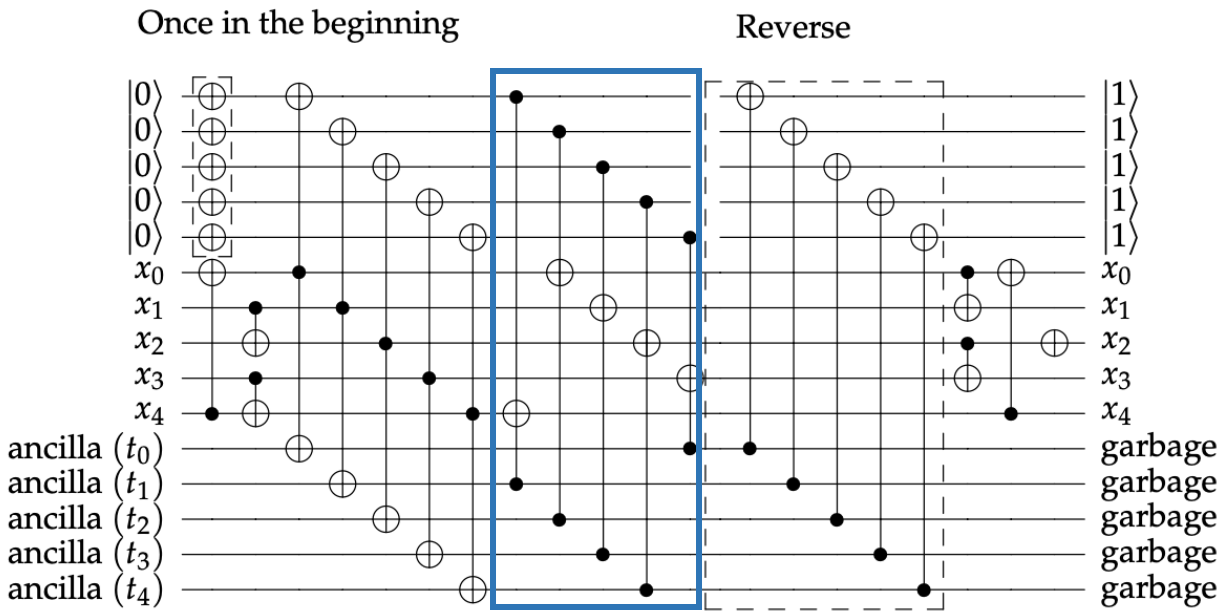
제안 기법

- **S-box 병렬 구현**
  - **모든 Toffoli gate 연산을 병렬로 처리 → Toffoli depth 1**
  - 두 세트의 보조 큐비트 할당 (**640 = 320 x 2**)

parallelization



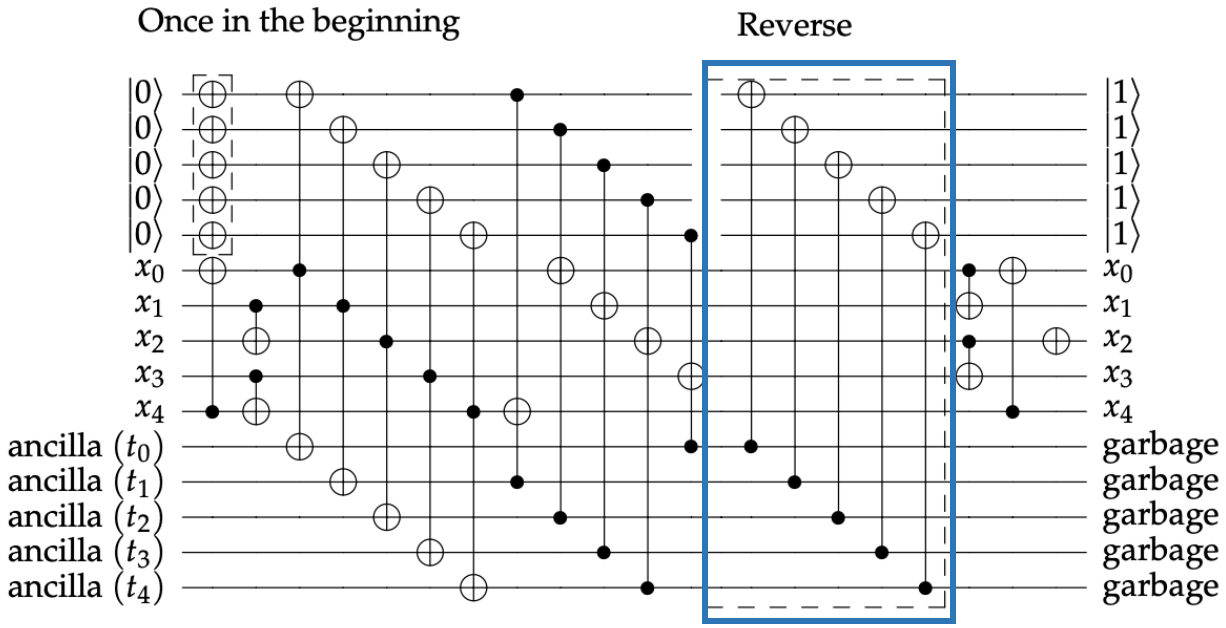
parallelization



제안 기법

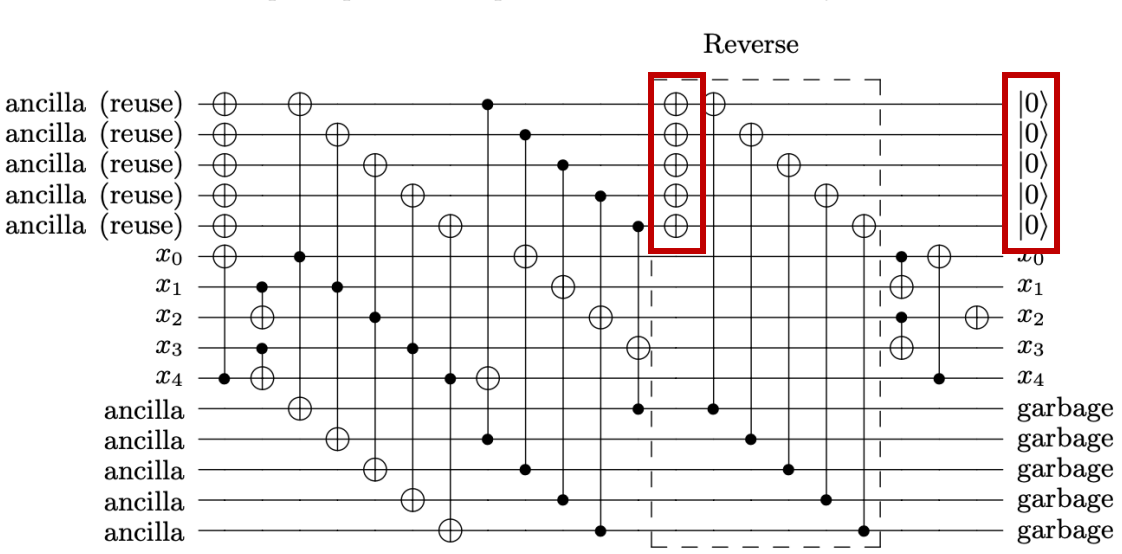
- **역연산을 통한 보조 큐비트 재사용**
  - **역연산**을 통해 하나의 **보조 큐비트 세트를 재사용.**

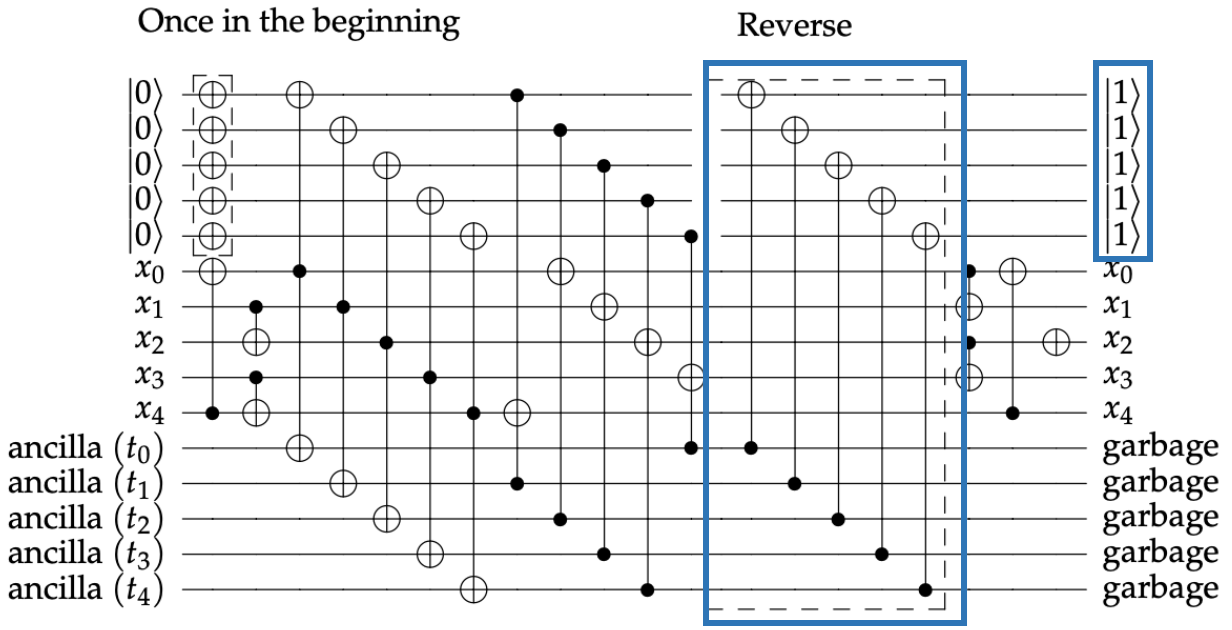
    → 초기 320개의 큐비트 할당만 필요



제안 기법

- **역연산을 통한 보조 큐비트 재사용**
  - 역연산 시 **NOT 연산을 하지 않음** → 보조 큐비트 상태를 $|1\rangle$
    → **다음 라운드부터는 NOT 연산이 필요 없음**





제안 기법

- **선형 레이어 구현**
  - Roy et al. 은 ASCON 선형 레이어 구현에 관한 다양한 기법을 연구하였음
  - Naïve한 구현 (즉, **out-of-place**)이 높은 큐비트 수를 요구하지만 **가장 낮은 depth**를 가짐
    - 모든 비트를 (320(=5 x 64)) 병렬로 처리함으로써 **depth 최적화**

Table 1: Comparison of quantum resources required for ASCON linear layer.

Linear layer	Source	#CNOT	#Qubit	Depth
Out-of-place	This work	960	640	3
Naïve (binary matrix)	RBC'23 [18]	960	640	26
Gauss-Jordan	RBC'23 [18]	2,413	320	358
PLU	RBC'23 [18]	2,413	320	288
Modified [19]	RBC'23 [18]	1,595	320	119

In-place

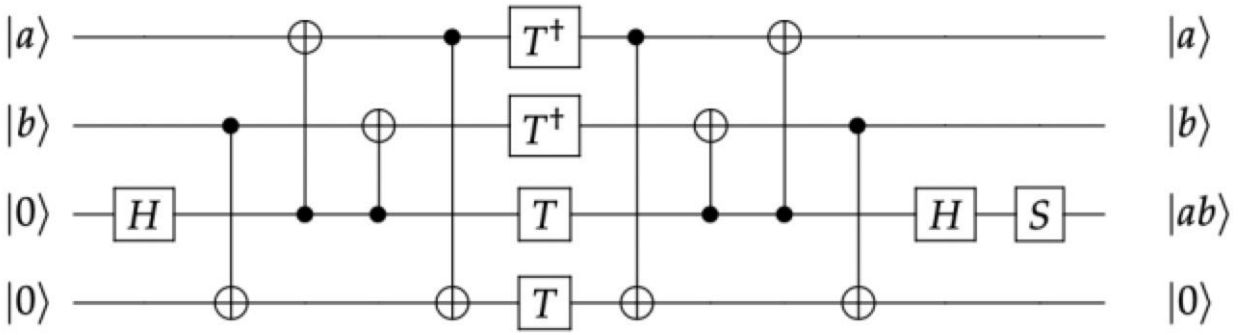
$$
\begin{aligned}
x_0 &\leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28),\\
x_1 &\leftarrow \Sigma_1(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39),\\
x_2 &\leftarrow \Sigma_2(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6),\\
x_3 &\leftarrow \Sigma_3(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17),\\
x_4 &\leftarrow \Sigma_4(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41),
\end{aligned}
$$

[수식 3-2] ASCON 선형 레이어

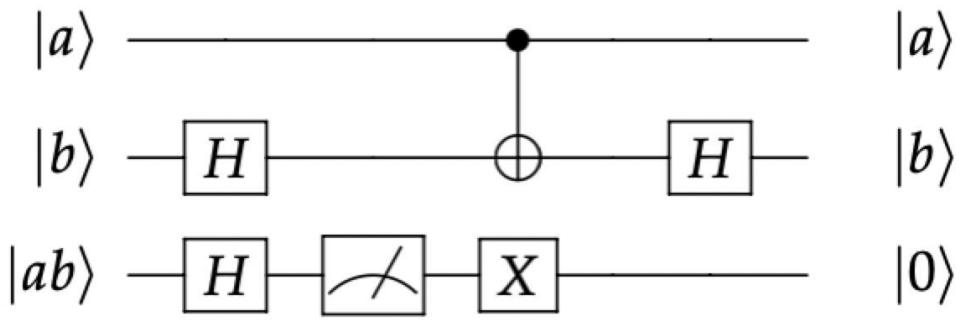
Roy, A. Baksi, and A. Chattopadhyay, "Quantum implementation of ascon linear layer," Cryptology ePrint Archive, 2023. 9

제안 기법

- **AND 게이트 사용**
  - Toffoli gate와 유사하게 동작, But **대상 큐비트가 clean (즉 0) 상태여야 함**
  - AND 게이트 : 11개의 Clifford 게이트, 4개의 T 게이트, 1개의 보조 큐비트로 구성 (T-depth 1, 전체 depth 8)
  - $\mathrm{AND}^\dagger$ 게이트 : 5개의 Clifford 게이트, 1개의 Measurement 게이트 (T-depth 0, 전체 depth 4)



[그림 3-2] AND 게이트 양자 회로



[그림 3-3] AND† 게이트 양자 회로

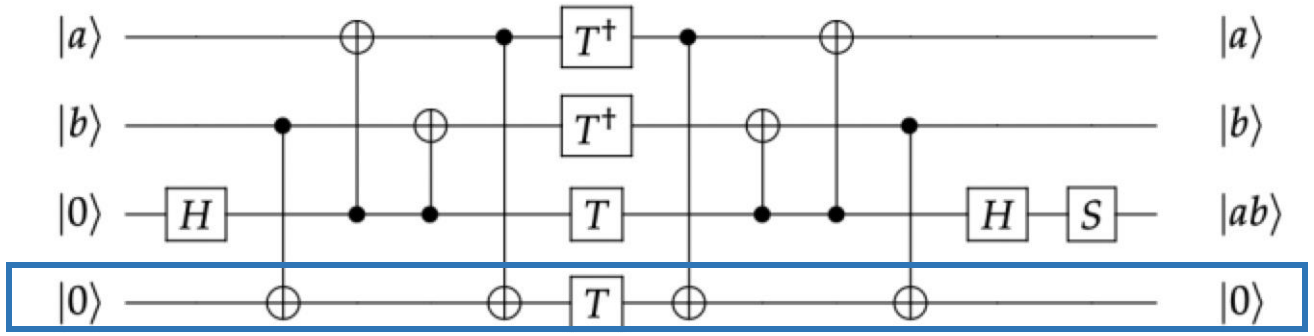
제안 기법

- **AND 게이트 사용**
  - AND 게이트 보조 큐비트 재사용 가능
  - 치환레이어에서 **320개 ($= 5 \times 64$) 보조 큐비트 초기에 한번만 할당**

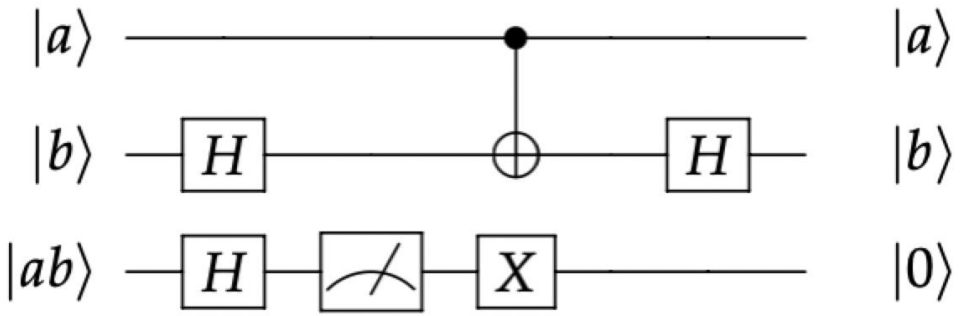
    → 할당 x, 선형 레이어에서 사용할 보조 큐비트를 미리 선언하여 사용
  - 본 구현에서 Toffoli 게이트 역연산이 사용 되지 않음

    → $\mathrm{AND}^{\dagger}$ 게이트의 자원 효율성 활용 못함

    → Grover 오라클에서 $\mathrm{AND}^{\dagger}$ 활용



[그림 3-2] AND 게이트 양자 회로



[그림 3-3] $\mathrm{AND}^{\dagger}$ 게이트 양자 회로

성능 평가

성능 평가

- **ASCON 양자 회로 자원 추정**
  - 본 구현은 낮은 Toffoli 및 Full depth 달성, 그러나 높은 큐비트 수

    → Trade off 메트릭인 $TD-M, FD-M\ TD^2-M, FD^2-M$도 구함

    → ASCON –HASH(256)의 경우 이전 연구와 비교하여 **모든 depth 및 trade-off 메트릭에서 최적화**

Cipher		Source	#CNOT	#1qCliff	#T	Toffoli Depth (TD)	#Qubit (M)	Full Depth (FD)	TD−M	FD−M	TD^2-M	FD^2-M
ASCON -AEAD	ASCON -128	Ours	127,200	21,563	67,220	30	20,064	513	1.15×2^{19}	1.23×2^{23}	1.08×2^{24}	1.23×2^{32}
	ASCON -128a	Ours	135,648	22,979	71,680	32	21,344	547	1.30×2^{19}	1.39×2^{23}	1.30×2^{24}	1.49×2^{32}
ASCON hash function	ASCON -HASH (256)	Lee	491,008	208,018	387,072	864	35,136	8,427	1.81×2^{24}	1.10×2^{28}	1.53×2^{34}	1.13×2^{41}
		Ours	406,016	68,435	215,040	96	62,592	1,641	1.43×2^{22}	1.53×2^{26}	1.07×2^{29}	1.23×2^{37}
	ASCON -XoF (384)	Ours	609,024	102,419	322,560	144	93,568	2,461	1.61×2^{23}	1.72×2^{27}	1.81×2^{30}	1.03×2^{39}
	ASCON -XoF (512)	Ours	812,032	136,402	430,080	192	124,544	3,281	1.43×2^{24}	1.52×2^{28}	1.07×2^{32}	1.22×2^{40}

[표 4-1] ASCON 양자 회로 구현에 사용된 양자 자원 비용 비교

성능 평가

- **Grover 공격 비용 추정**
  - Grover 오라클 비용 : **양자 자원 비용 $\times$ 2**
    - AND 게이트 사용 : AND 게이트 양자 자원 비용 + $\mathrm{AND}^{\dagger}$ 양자 자원 비용
  - Grover 키 검색 비용 : **오라클 비용** $\times \left\lfloor \frac{\pi}{4}\sqrt{2^k} \right\rfloor$

Cipher		Source	#CNOT	#1qCliff	#T	#Measure	T Depth (Td)	#Qubit (M)	Full Depth (FD)	Td−M	FD−M	Td^2-M	FD^2-M
ASCON-AEAD	ASCON-128	Ours	254,400	43,126	134,440	0	240	20,065	1,026	1.15×2^{22}	1.23×2^{24}	1.08×2^{30}	1.23×2^{34}
		Ours-AND	225,600	71,926	38,400	9,600	30	20,065	816	1.15×2^{19}	1.95×2^{23}	1.08×2^{24}	1.56×2^{33}
	ASCON-128a	Ours	271,296	45,958	143,360	0	256	21,355	1,094	1.30×2^{22}	1.40×2^{24}	1.30×2^{30}	1.49×2^{34}
		Ours-AND	240,576	76,678	40,960	10,240	32	21,355	872	1.30×2^{19}	1.11×2^{24}	1.30×2^{24}	1.89×2^{33}

ASCON-AEAD에 대한 Grover 오라클 비용

Cipher		Source	#Gate (G)	Full Depth (FD)	T Depth (Td)	#Qubit (M)	G−FD	FD−M	Td−M	FD^2-M	Td^2-M
ASCON-AEAD	ASCON-128	Ours	1.31×2^{82}	1.57×2^{73}	1.47×2^{71}	1.22×2^{14}	1.03×2^{156}	1.92×2^{87}	1.79×2^{85}	1.50×2^{161}	1.32×2^{157}
		Ours-AND	1.01×2^{82}	1.25×2^{73}	1.44×2^{68}	1.22×2^{14}	1.26×2^{155}	1.53×2^{87}	1.76×2^{82}	1.90×2^{160}	1.27×2^{151}
	ASCON-128a	Ours	1.39×2^{82}	1.68×2^{73}	1.57×2^{71}	1.30×2^{14}	1.17×2^{156}	1.10×2^{88}	1.02×2^{86}	1.83×2^{161}	1.60×2^{157}
		Ours-AND	1.10×2^{82}	1.34×2^{73}	1.56×2^{68}	1.30×2^{14}	1.47×2^{155}	1.74×2^{87}	1.01×2^{83}	1.17×2^{161}	1.58×2^{151}

[표 4-3] ASCON-AEAD에 대한 Grover 공격 비용

성능 평가

- **Grover 공격 비용 추정**
  - Grover 양자 충돌 공격 : CNS 알고리즘 적용 $O(2^{\frac{2n}{5}-\frac{3s}{5}})$ $(s \leq \frac{n}{4})$.
  - Grover 오라클 비용 : **양자 자원 비용 $\times$ 2**
    - AND 게이트 사용 : AND 게이트 양자 자원 비용 + $\text{AND}^{\dagger}$ 양자 자원 비용
  - Grover 양자 충돌 공격 비용 : **오라클 비용** $\times\ 2^{(\frac{2n}{5}-\frac{3s}{5})}$ $s = \frac{n}{6}$

ASCON hash function	ASCON-HASH (256)	Lee	982,016	416,036	774,144	0	6,912	35,137	16,854	1.81×2^{25}	1.10×2^{29}	1.53×2^{36}	1.13×2^{43}
		Ours	812,032	136,870	430,080	0	768	62,593	3,282	1.43×2^{25}	1.53×2^{27}	1.07×2^{35}	1.23×2^{39}
		Ours-AND	719,872	229,030	122,880	30,720	96	62,593	2,608	1.43×2^{22}	1.22×2^{27}	1.07×2^{29}	1.55×2^{38}
	ASCON-XoF (384)	Ours	1,218,048	204,838	645,120	0	1,152	93,569	4,922	1.61×2^{26}	1.72×2^{28}	1.81×2^{36}	1.03×2^{41}
		Ours-AND	1,079,808	343,076	184,320	46,080	144	93,569	3,904	1.61×2^{23}	1.36×2^{28}	1.81×2^{30}	1.55×2^{38}
	ASCON-XoF (512)	Ours	1,624,064	272,804	860,160	0	1,536	124,545	6,562	1.43×2^{27}	1.52×2^{29}	1.07×2^{38}	1.22×2^{42}
		Ours-AND	1,439,744	457,124	245,760	61,440	192	124,545	5,200	1.43×2^{24}	1.21×2^{29}	1.07×2^{32}	1.53×2^{41}

ASCON 해시함수에 대한 Grover 오라클 비용

Cipher		Source	#Gate (G)	Full Depth (FD)	T Depth (Td)	#Qubit (M)	G−FD	FD−M	Td−M	FD^2-M	Td^2-M
ASCON hash function	ASCON-HASH	Lee	1.42×2^{97}	1.40×2^{90}	1.15×2^{89}	1.70×2^{57}	1.99×2^{187}	1.19×2^{148}	1.96×2^{146}	1.68×2^{238}	1.13×2^{236}
		Ours	1.80×2^{96}	1.09×2^{88}	1.02×2^{86}	1.51×2^{58}	1.96×2^{184}	1.66×2^{146}	1.55×2^{144}	1.81×2^{234}	1.59×2^{230}
		Ours-AND	1.44×2^{96}	1.74×2^{87}	1.02×2^{83}	1.51×2^{58}	1.25×2^{184}	1.25×2^{146}	1.54×2^{141}	1.14×2^{234}	1.57×2^{224}
	ASCON-XoF (384)	Ours	1.78×2^{135}	1.08×2^{127}	1.01×2^{125}	1.42×2^{80}	1.92×2^{262}	1.54×2^{207}	1.44×2^{205}	1.67×2^{334}	1.46×2^{330}
		Ours-AND	1.42×2^{135}	1.67×2^{126}	1.01×2^{122}	1.42×2^{80}	1.19×2^{262}	1.19×2^{207}	1.44×2^{202}	1.00×2^{334}	1.46×2^{324}
	ASCON-XoF (512)	Ours	1.57×2^{174}	1.90×2^{165}	1.78×2^{163}	1.19×2^{102}	1.49×2^{340}	1.14×2^{268}	1.06×2^{266}	1.08×2^{434}	1.90×2^{429}
		Ours-AND	1.25×2^{174}	1.51×2^{165}	1.77×2^{160}	1.19×2^{102}	1.89×2^{339}	1.80×2^{267}	1.06×2^{263}	1.36×2^{433}	1.88×2^{423}

[표 4-4] ASCON 해시함수에 대한 Grover 양자 충돌 공격 비용

결 론

결론

- ASCON-128 : 1.26×2^{155}. ASCON-128a : 1.47×2^{155}
  - **보안 레벨 1을 만족하지 못함**
- ASCON-HASH (256) : 1.25×2^{184}. ASCON-XoF (384) : 1.19×2^{262}. ASCON-HASH (512) : 1.89×2^{339}
  - **Jang et al. 이 정의한 보안레벨 중 SHA3 공격 비용 만족**

보안 레벨	암호	비용(복잡도)
Level 1	AES-128	$2^{170} \rightarrow 2^{157}$
Level 2	SHA-256/SHA3-256	2^{146} (classical gates)
Level 3	AES-192	$2^{233} \rightarrow 2^{221}$
Level 4	SHA-384/SHA3-384	2^{210} (classical gates)
Level 5	AES-256	$2^{198} \rightarrow 2^{285}$

[표 2-3] NIST 양자 후 보안 레벨

Level	Cipher	Cost
Level 2	SHA-2/3 (256)	$2^{188}/2^{183}$
Level 4	SHA-2/3 (384)	$2^{266}/2^{260}$
Level 6 (Extension)	SHA-2/3 (512)	$2^{343}/2^{337}$

[표 2-4] Jang et al.이 정의한 양자 충돌 공격에 대한 보안 레벨

Jang, K., Lim, S., Oh, Y., Kim, H., Baksi, A., Chakraborty, S., & Seo, H. (2024). Quantum Implementation and Analysis of SHA-2 and SHA-3. Cryptology ePrint Archive.

결론

- 본 논문에서는 ASCON-AEAD 및 해시함수 양자 회로 최적화 구현하고 공격 비용을 평가함
- ASCON-128, ASCON-128a는 NIST에서 제공하는 **보안 레벨 1을 달성하지 못함**
  - 키 길이 증가 필요
- ASCON-HASH (256비트)의 경우 **Toffoli depth 88.9%, Full depth : 80.5%** 이상 향상됨
- ASCON-HASH (256), ASCON-XoF(384, 512)는 Jang et al. 이 **SHA3에 대한 공격 비용 만족**
- ASCON 저자들은 추가로 양자 공격에 저항을 갖는 160비트 더 긴 ASCON-80pq 제안
- 향후 ASCON-80pq에 대한 공격 비용도 평가 예정

Q & A