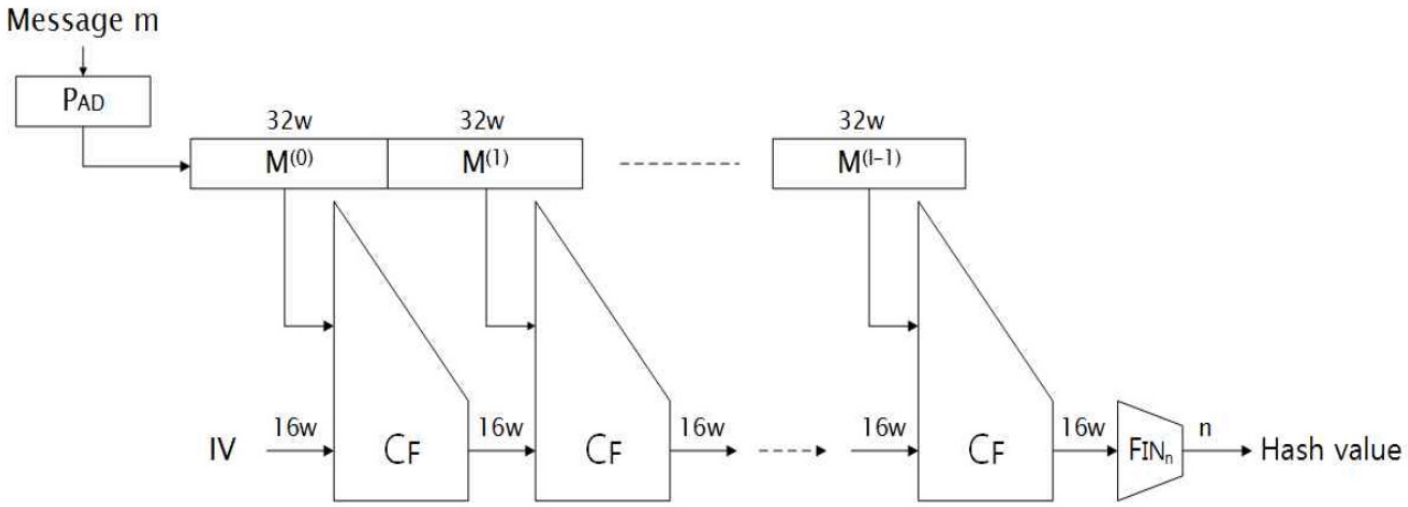
LSH on Quantum

https://youtu.be/lDhTATTGGaE

LSH

- LSH 구조
  - Initialization
    - 입력 메시지 패딩, IV 값을 CV에 저장
  - Compression
    - MsgExp
    - MsgAdd
    - Mix
    - WordPerm

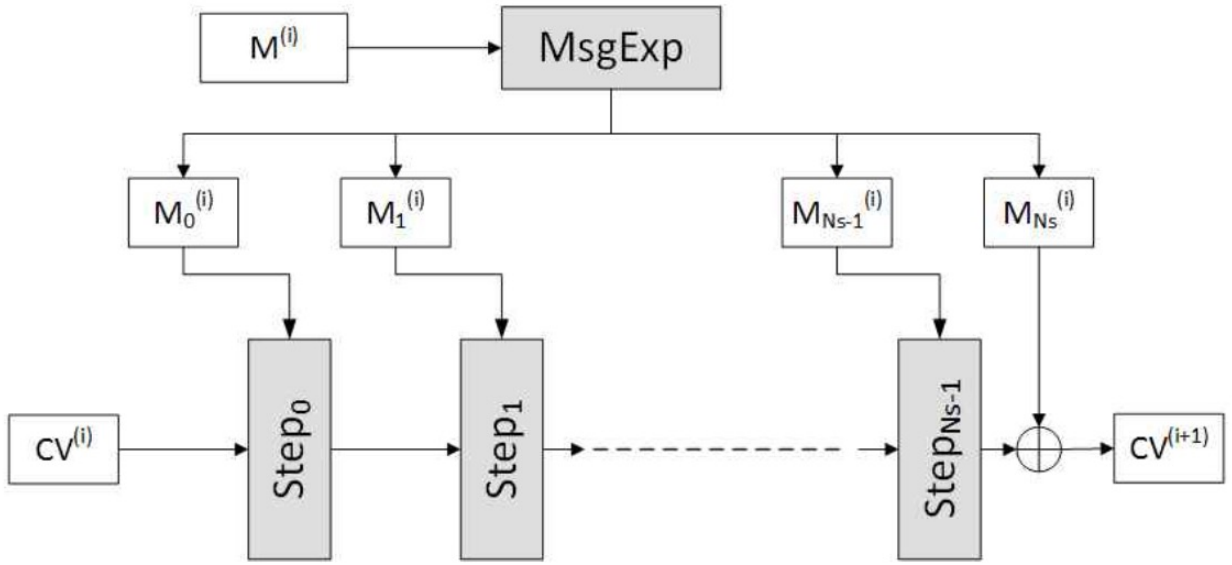
    ] Step 함수
  - Finalization
    - n 비트 길이의 해시 값 h를 생성



LSH

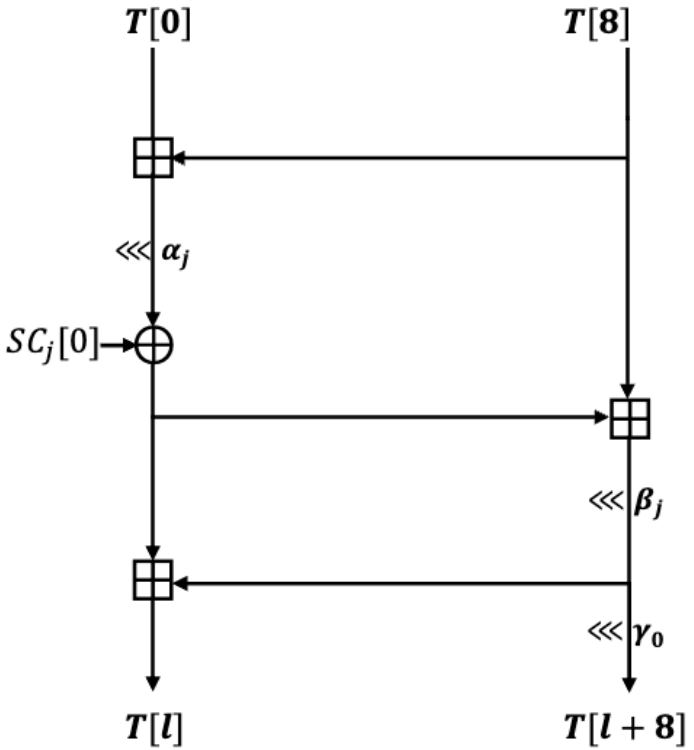
- Compression
  - MsgExp : 메시지 확장 함수
    - 32 워드 배열 → 16 워드 배열
    - $\mathrm{M}^{(i)} = (\mathrm{M}^{(i)}[0], \ldots, \mathrm{M}^{(i)}[31])$

$$\mathbf{M}_0^{(i)} \leftarrow (M^{(i)}[0], \ldots, M^{(i)}[15]),\ \mathbf{M}_1^{(i)} \leftarrow (M^{(i)}[16], \ldots, M^{(i)}[31])$$
$$\mathbf{M}_j^{(i)} \leftarrow (M_j^{(i)}[0], \ldots, M_j^{(i)}[15])_{j=2}^{N_s}$$
$$M_j^{(i)}[l] \leftarrow M_{j-1}^{(i)}[l] \boxplus M_{j-2}^{(i)}[\tau(l)]\ for\ 0 \le l \le 16$$



LSH

- Step
  - MsgADD
    - msg와 CV 값 xor 연산
    - $MsgADD(X, Y) = (X[0] \oplus Y[0], \cdots X[15] \oplus Y[15])$
  - Mix
    - $(T[l], T[l\ +\ 8]) \leftarrow\ Mix_{j,l}\ (T[l], T[l\ +\ 8])\ for\ 0 \leq l < 8$
    - 3개의 덧셈과 xor, rotation 사용
  - WordPerm
    - $WordPerm(X) = X[\sigma(0)], \cdots, X[\sigma(15)]$



Mix 함수

LSH on Quantum

- LSH 양자 회로
  - Initialization
    → not 연산만을 사용 (IV constant값을 CV 큐비트에 저장, 패딩)
  - Compression
    → **add**, rotation, not (SC constant xor) 연산 사용
    → rotation 연산은 logical 로 구현 → 자원 추정 x
  - Finalization
    → xor 연산 사용

덧셈기와 병렬처리가 최적화하는데 중요함

LSH on Quantum

- **Draper adder**
  - Carry-lookahead adder(CLA)
    - 각 자리수의 캐리를 미리 계산

    → 이전 자리의 캐리 값이 구해질 때까지(RCA) 기다리지 않아도 됨.

    →많은 큐비트 수, 낮은 depth

Table 1: Comparison of quantum resources required for adder (32-bit).

Adder	Operation	#CNOT	#Toffoli	Toffoli depth	#Qubit (reuse)	Depth
Cuccaro [2]	in-place	153	61	61	65	66
Draper [3]	in-place	123	254	22	117 (53)	28
	out-of-place	94	127	11	118 (22)	14

※: Estimation of undecomposed resources

LSH on Quantum

- **Parallel in Addition**
  - MsgExp
    - on-the-fly 형태로 연산, 16개의 덧셈이 사용

$$\mathbf{M}_0^{(i)} \leftarrow (M^{(i)}[0], ..., M^{(i)}[15]), \mathbf{M}_1^{(i)} \leftarrow (M^{(i)}[16], ..., M^{(i)}[31])$$
$$\mathbf{M}_j^{(i)} \leftarrow (M_j^{(i)}[0], ..., M_j^{(i)}[15])_{j=2}^{N_s}$$
$$M_j^{(i)}[l] \leftarrow M_{j-1}^{(i)}[l] \boxplus M_{j-2}^{(i)}[\tau(l)] \; for \; 0 \leq l \leq 16$$

    - 순차적인 덧셈 연산

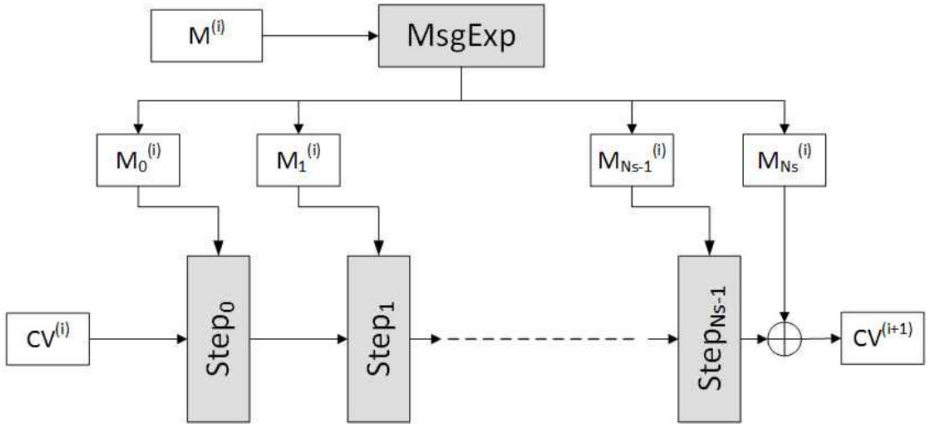
      →초기에 53개의 안실라 큐비트 할당만 필요,

      →높은 depth

    - **병렬 덧셈 연산**

      → 초기에 **848 (16 × 53)**개의 안실라 큐비트 할당

      → **낮은 depth**



LSH on Quantum

- **Parallel in Addition**
  - Mix
    - **24 (8 × 3)** 개의 덧셈 사용
    - $(T[l], T[l+8]) \leftarrow Mix_{j,l}(T[l], T[l+8])$
    - 8개씩 병렬 덧셈 가능

      → **424 (8 × 53)** 개의 안실라 큐비트 필요

      → MsgExp에서 사용한 안실라 큐비트 **재사용 가능**

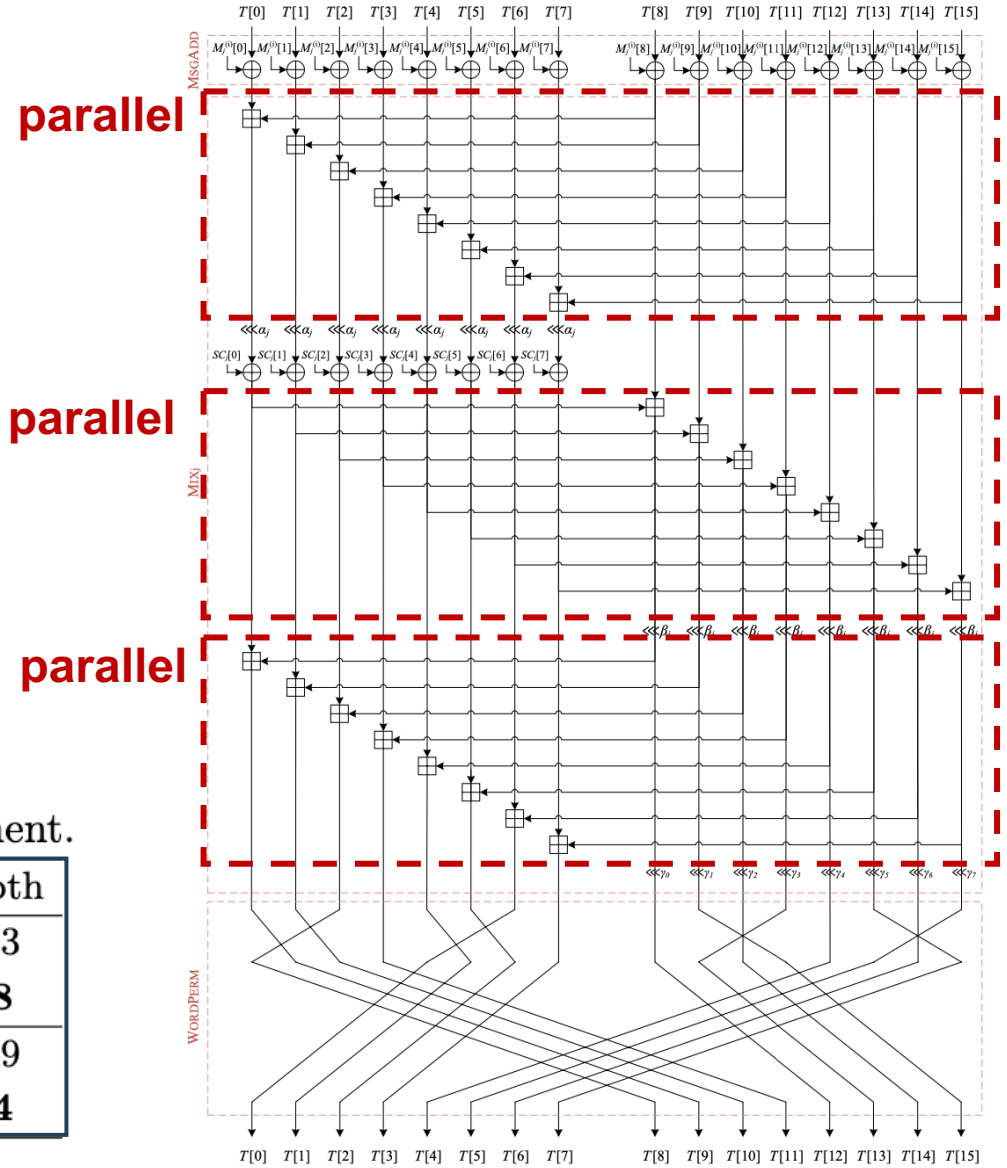


Table 2: Comparison of quantum resources required for each component.

Function	Operation	#CNOT	#Toffoli	Toffoli depth	#Qubit	Depth
MsgExp	Sequential	1,968	4,064	352	1,077	433
	Parallel	1,968	4,064	**22**	1,872	**28**
Mix	Sequential	2,952	6,096	528	565	649
	Parallel	2,952	6,096	**66**	936	**84**

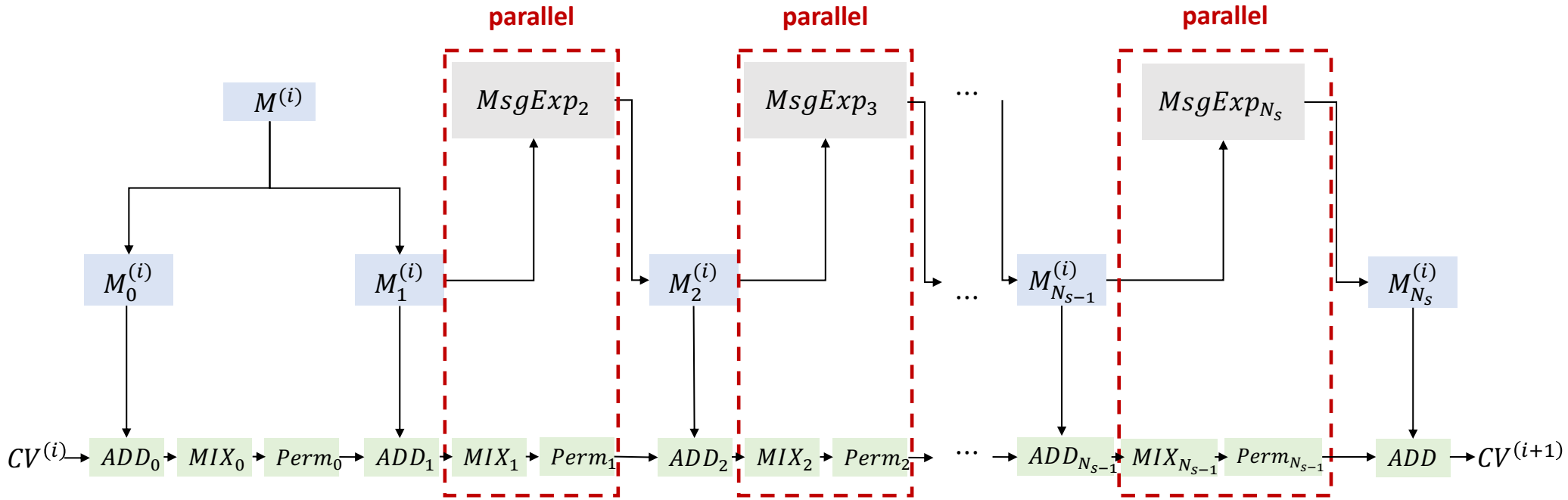
※: Estimation of undecomposed resources

LSH on Quantum

- **Parallel in Compression**
  - MsgExp와 Mix 함수는 각각 **독립적**으로 수행 가능
    - 정확하게는 **i번째 Mix**와 **$i+1$번째 MsgExp**이 독립적 → **병렬 연산 가능**

      → But, Mix 함수에서 MsgExp에서 사용한 안실라 재사용할 경우 불가능

      → **Mix 와 MsgExp에서 사용할 안실라 큐비트를 각각 할당**



LSH on Quantum

- **Parallel in Compression**

Algorithm 1: Quantum circuit implementation.

Input: M_{even},M_{odd} CV, α, β, SC, $ancilla_0$, $ancilla_1$
Output: M_{even}, M_{odd}, CV, $ancilla_0$, $ancilla_1$

1: $CV \leftarrow \text{MsgAdd}(M_{even}, CV)$
2: $CV \leftarrow \text{Mix}(CV, \alpha_{even}, \beta_{even}, SC, ancilla_0)$
3: $CV \leftarrow \text{WordPerm}(CV)$

4: $CV \leftarrow \text{MsgAdd}(M_{odd}, CV)$
5: $CV \leftarrow \text{Mix}(CV, \alpha_{odd}, \beta_{odd}, SC, ancilla_0)$
6: $CV \leftarrow \text{WordPerm}(CV)$

7: **for** $1 \leq i \leq 13$ **do**
8: $M_{even} \leftarrow \text{MsgExp}(M_{even}, M_{odd}, ancilla_1)$
9: $CV \leftarrow \text{MsgAdd}(M_{even}, CV)$
10: $CV \leftarrow \text{Mix}(CV, \alpha_{even}, \beta_{even}, SC, ancilla_0)$
11: $CV \leftarrow \text{WordPerm}(CV)$

12: $M_{odd} \leftarrow \text{MsgExp}(M_{even}, M_{odd}, ancilla_1)$
13: $CV \leftarrow \text{MsgAdd}(M_{odd}, CV)$
14: $CV \leftarrow \text{Mix}(CV, \alpha_{odd}, \beta_{odd}, SC, ancilla_0)$
15: $CV \leftarrow \text{WordPerm}(CV)$
16: **end for**

17: $M_{even} \leftarrow \text{MsgExp}(M_{even}, M_{odd}, ancilla_1)$
18: $CV \leftarrow \text{MsgAdd}(M_{even}, CV)$

19: return CV

- $\boldsymbol{ancilla_0}$
  - Mix에 사용되는 **424 (8 × 53)**개의 안실라 큐비트
- $\boldsymbol{ancilla_1}$
  - MsgExp에 사용되는 **848 (16 × 53)**개의 안실라 큐비트

- Mix 함수는 병렬 덧셈이 3번 반복됨
  → **MsgExp의 depth는 생략되고 Mix의 depth가 추정됨**

Table 3: Comparison of quantum resources required for the Compression function.

Function	Operation	#CNOT	#Toffoli	Toffoli depth	#Qubit	Depth
Compression	Sequential	139,776	260,096	2,266	2,384	2,873
	Parallel	139,776	260,096	**1,716**	2,808	**2,198**

※: Estimation of undecomposed resources

Evaluation

- **Evaluation**
  - 이전 연구, Ours-CDKM(이전 연구 + 병렬 최적화), Ours (병렬 + Draper) 비교
  - Ours가 게이트 수와 큐비트는 가장 많이 사용
  - **TD, FD 가장 좋은 성능. + trade-off 메트릭 모두에서 더 좋은 성능을 보임**
  - Grover 공격 비용도 G-FD는 Ours-CDKM이 조금 더 좋은 성능이지만 다른 메트릭 Depth 관련 메트릭은 Ours가 가장 좋은 성능을 보임

Table 4: Quantum resources required for implementations of LSH.

Cipher	Source	#CNOT	#1qCliff	#T	Toffoli depth (TD)	#Qubit (M)	Full depth (FD)	TD-M	FD-M	TD^2-M	FD^2-M
LSH-256-256	[12]	545,536	187,813	437,248	6,283	1,552	50,758	$1.16 \cdot 2^{23}$	$1.17 \cdot 2^{26}$	$1.78 \cdot 2^{35}$	$1.82 \cdot 2^{41}$
	Ours-CDKM	545,536	187813	437,248	**4,758**	1,560	**38,483**	$\mathbf{1.77 \cdot 2^{22}}$	$\mathbf{1.79 \cdot 2^{25}}$	$\mathbf{1.03 \cdot 2^{35}}$	$\mathbf{1.05 \cdot 2^{41}}$
	Ours	1,700,608	306,947	1,820,672	**1,716**	2,808	**13,647**	$\mathbf{1.15 \cdot 2^{22}}$	$\mathbf{1.14 \cdot 2^{25}}$	$\mathbf{1.93 \cdot 2^{32}}$	$\mathbf{1.90 \cdot 2^{38}}$
LSH-512-512	[12]	1,203,760	418,369	966,000	13,875	3,088	111,532	$1.28 \cdot 2^{25}$	$1.28 \cdot 2^{28}$	$1.08 \cdot 2^{39}$	$1.09 \cdot 2^{45}$
	Ours-CDKM	1,203,760	418,369	966,000	**10,500**	3,096	**84,451**	$\mathbf{1.94 \cdot 2^{24}}$	$\mathbf{1.95 \cdot 2^{27}}$	$\mathbf{1.24 \cdot 2^{38}}$	$\mathbf{1.26 \cdot 2^{44}}$
	Ours	4,030,000	736,569	2,614,473	**2,028**	5,832	**17,385**	$\mathbf{1.41 \cdot 2^{23}}$	$\mathbf{1.51 \cdot 2^{26}}$	$\mathbf{1.40 \cdot 2^{34}}$	$\mathbf{1.60 \cdot 2^{40}}$

Table 5: Costs of the Grover's collision search for LSH.

Cipher	Source	#Gate (G)	Full depth (FD)	T-depth (Td)	#Qubit (M)	G-FD	FD-M	Td-M	FD^2-M	Td^2-M
LSH-256-256	[12]	$1.35 \cdot 2^{107}$	$1.53 \cdot 2^{101}$	$1.51 \cdot 2^{100}$	$1.51 \cdot 2^{10}$	$1.03 \cdot 2^{209}$	$1.16 \cdot 2^{112}$	$1.15 \cdot 2^{111}$	$1.78 \cdot 2^{213}$	$1.74 \cdot 2^{211}$
	Ours-CDKM	$1.10 \cdot 2^{106}$	$1.16 \cdot 2^{101}$	$1.14 \cdot 2^{100}$	$1.52 \cdot 2^{10}$	$\mathbf{1.28 \cdot 2^{207}}$	$1.77 \cdot 2^{111}$	$1.75 \cdot 2^{110}$	$1.02 \cdot 2^{213}$	$1.00 \cdot 2^{211}$
	Ours	$1.81 \cdot 2^{107}$	$1.64 \cdot 2^{99}$	$1.65 \cdot 2^{98}$	$1.37 \cdot 2^{11}$	$\mathbf{1.48 \cdot 2^{207}}$	$1.13 \cdot 2^{111}$	$1.13 \cdot 2^{110}$	$1.86 \cdot 2^{210}$	$1.88 \cdot 2^{208}$
LSH-512-512	[12]	$1.22 \cdot 2^{107}$	$1.68 \cdot 2^{102}$	$1.67 \cdot 2^{101}$	$1.51 \cdot 2^{11}$	$1.02 \cdot 2^{210}$	$1.27 \cdot 2^{114}$	$1.26 \cdot 2^{113}$	$1.06 \cdot 2^{217}$	$1.05 \cdot 2^{215}$
	Ours-CDKM	$1.22 \cdot 2^{107}$	$1.27 \cdot 2^{102}$	$1.26 \cdot 2^{101}$	$1.02 \cdot 2^{11}$	$\mathbf{1.55 \cdot 2^{209}}$	$1.92 \cdot 2^{113}$	$1.91 \cdot 2^{112}$	$1.22 \cdot 2^{216}$	$1.21 \cdot 2^{214}$
	Ours	$1.74 \cdot 2^{108}$	$1.04 \cdot 2^{100}$	$1.95 \cdot 2^{98}$	$1.42 \cdot 2^{12}$	$\mathbf{1.82 \cdot 2^{208}}$	$1.49 \cdot 2^{112}$	$1.39 \cdot 2^{111}$	$1.56 \cdot 2^{212}$	$1.36 \cdot 2^{210}$

Q & A