# Quantum Collision Search on Hash Functions (the CNS Algorithm)

장경배

https://youtu.be/4xzw-VC1Fjw

HANSUNG UNIVERSITY

CryptoCraft LAB

# Grover on Hash Functions

- **Pre-image attack**
  - **주어진 (known) 해시 값을 생성하는 input 값 (unknown)을 찾아내는 것**
    - Hash$(x)$ = Known-output

    - **$n$-bit이 known-output이 주어졌을 때, $n$-bit input을 대상으로 search**
      → 블록암호에 대한 key search와 유사

- **Collision search**
  - **다른 input 값이지만, 동일한 해시 값을 생성하는 쌍을 찾아내는 것**
    - Hash $(x_1)$ = Hash $(x_2)$

  - Pre-image attack과는 달리, 다양한 접근이 가능
    - **Second pre-image attack**
    - **BHT algorithm**

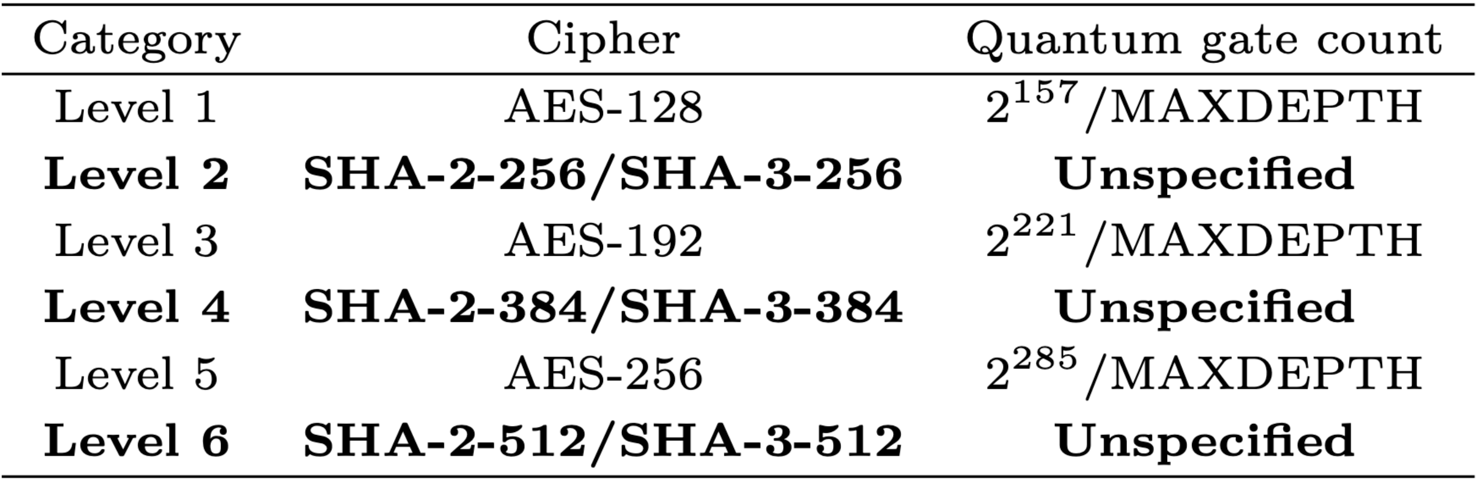
# Grover on Hash Functions

- **Second pre-image attack** (Quantum)
  - Input에 대한 output 해시가 주어졌을 때, **output을 생성하는 또 다른 input을 찾는 것**
    - Hash(Known-input ($n$-bit)) = Known-output ($n$-bit)
    - Hash($x$ ≠ Known-input) = Known-output
      → Quantum complexity: $\boldsymbol{O(2^{n/2})}$

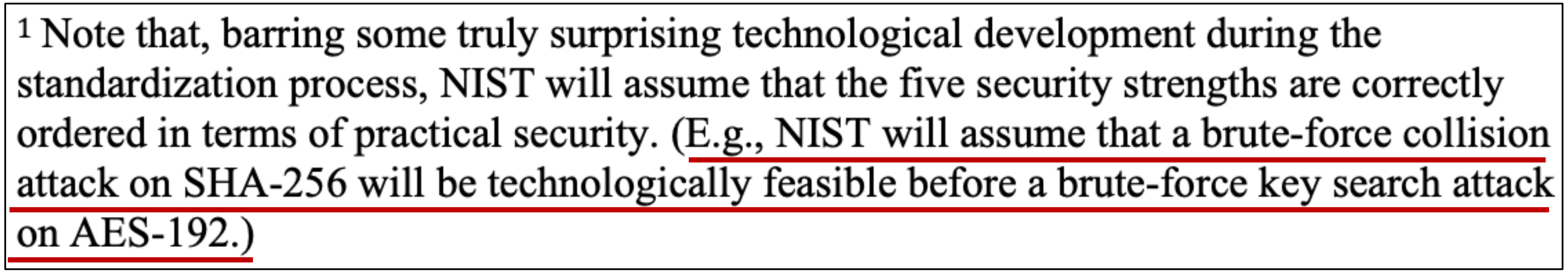
- 기본적인 방법이며, **복잡도는 Pre-image attack과 동일**함
- 간단하며, **Quantum ram이 필요 없다는 것이 장점**

# Grover on Hash Functions

- NIST의 post-quantum security level을 고려했을 때, **Second pre-image attack (Quantum, $O(n^{1/2})$) 은 적절하지 않음**



| Search Complexity | Category | Cipher | Quantum gate count |
|---|---|---|---|
| $2^{64}$ (key search) | Level 1 | AES-128 | $2^{157}$/MAXDEPTH |
| **$2^{128}$ (second pre-image)** | **Level 2** | **SHA-2-256/SHA-3-256** | **Unspecified** |
| $2^{96}$ (key search) | Level 3 | AES-192 | $2^{221}$/MAXDEPTH |
| **$2^{192}$ (second pre-image)** | **Level 4** | **SHA-2-384/SHA-3-384** | **Unspecified** |
| $2^{128}$ (key search) | Level 5 | AES-256 | $2^{285}$/MAXDEPTH |
| **$2^{256}$ (second pre-image)** | **Level 6** | **SHA-2-512/SHA-3-512** | **Unspecified** |



[1] Note that, barring some truly surprising technological development during the standardization process, NIST will assume that the five security strengths are correctly ordered in terms of practical security. (E.g., NIST will assume that a brute-force collision attack on SHA-256 will be technologically feasible before a brute-force key search attack on AES-192.)

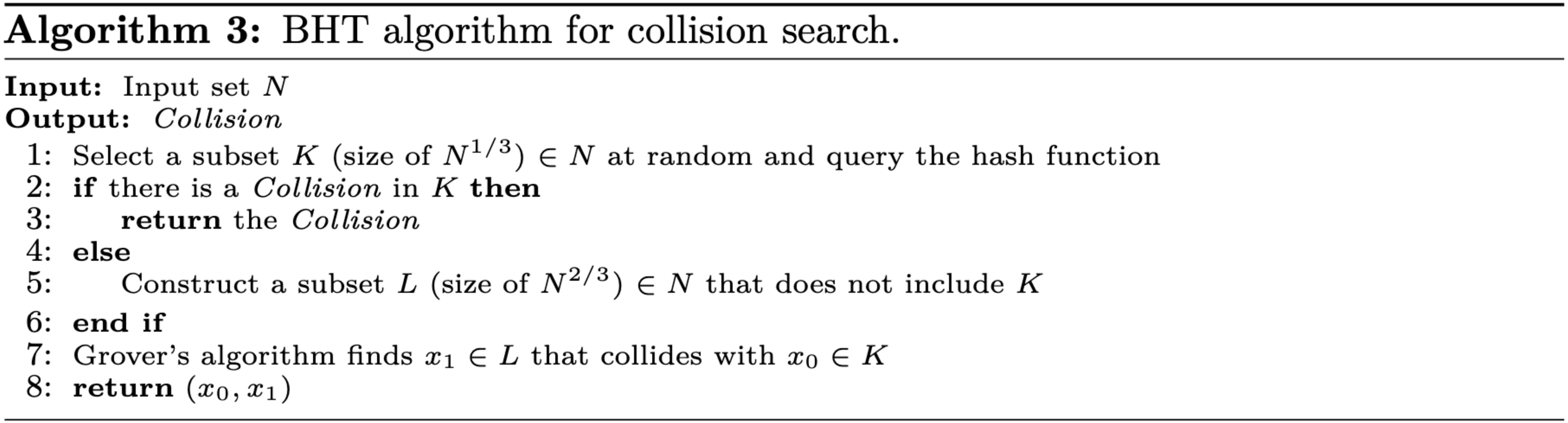
# Grover on Hash Functions

- **BHT algorithm**
  - Birthday paradox와 Grover's search를 결합한 알고리즘
    → Birthday paradox: 지정한 생일에 대한 확률은 낮지만, 같은 생일을 찾을 확률은 높음

1. **$2^{n/3}$의 무작위 input**으로 구성되는 **Subset $L$**을 구성
2. Subset $L$에서 collision이 발생하는지 확인 (Classical) → $\boldsymbol{O(2^{n/3})}$
   → Hash($x_0 \in K$) = Hash($x_1 \in L$), Go to step 5.
3. **Subset $L$을 제외한 input $2^{2n/3}$**으로 구성되는 **Subset $K$**를 구성
4. Grover's search는 Subset K ($2^{2n/3}$)에서 다음 솔루션을 찾음 → $\boldsymbol{O(2^{n/3})}$
   → Hash($x_0 \in K$) = Hash($x_1 \in L$)
5. return $(x_0, x_1)$

- **Quantum ram이 필요하다는 고려 사항이 있음 + 논쟁?의 여지가 있음**

# Grover on Hash Functions

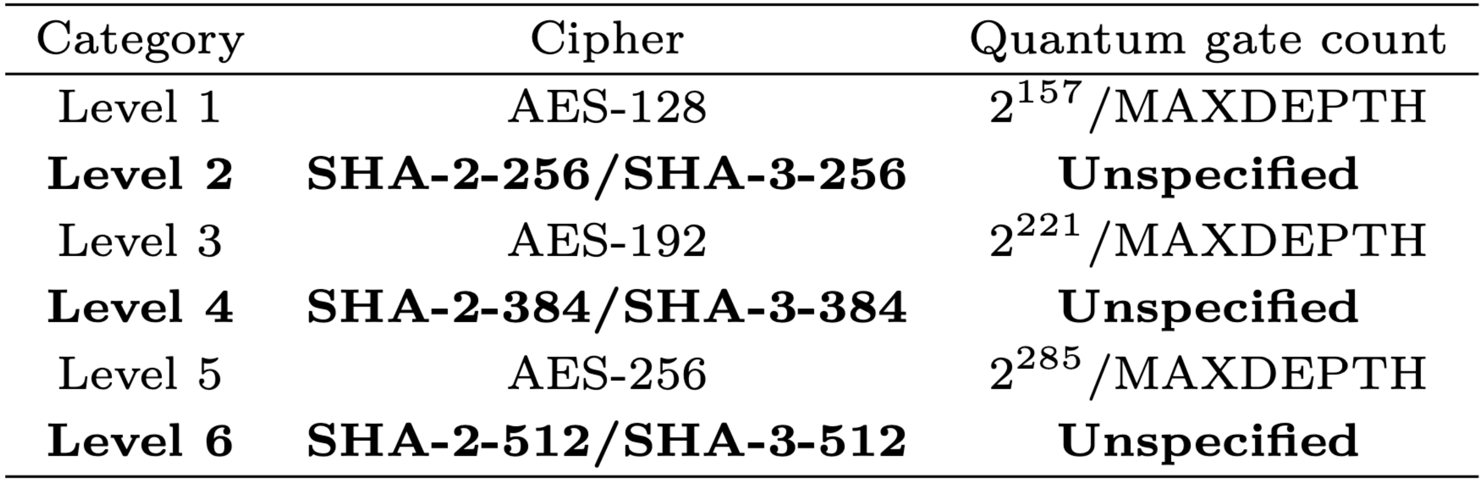


**Algorithm 3:** BHT algorithm for collision search.

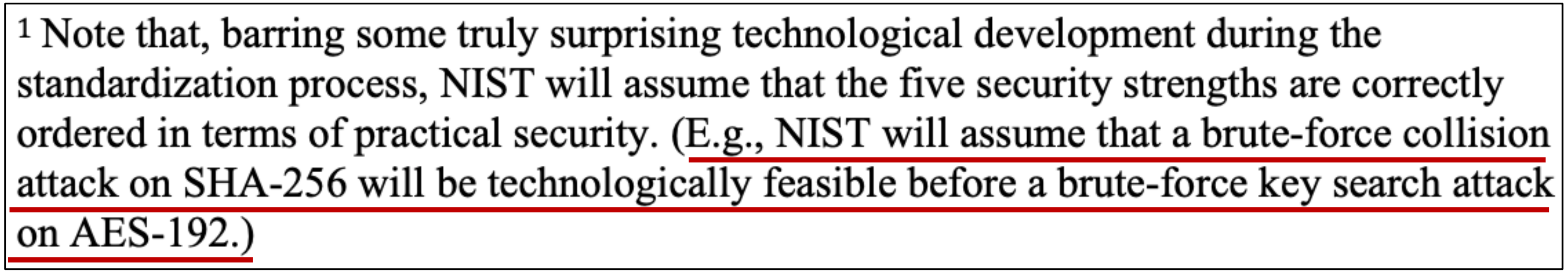
```
Input: Input set $N$
Output: $Collision$
Select a subset $K$ (size of $N^{1/3}$) $\in N$ at random and query the hash function
if there is a $Collision$ in $K$ then
    return the $Collision$
else
    Construct a subset $L$ (size of $N^{2/3}$) $\in N$ that does not include $K$
end if
Grover's algorithm finds $x_1 \in L$ that collides with $x_0 \in K$
return $(x_0, x_1)$
```

# Grover on Hash Functions

- NIST의 post-quantum security level을 고려했을 때, **BHT 알고리즘은 적절할 수 있음**



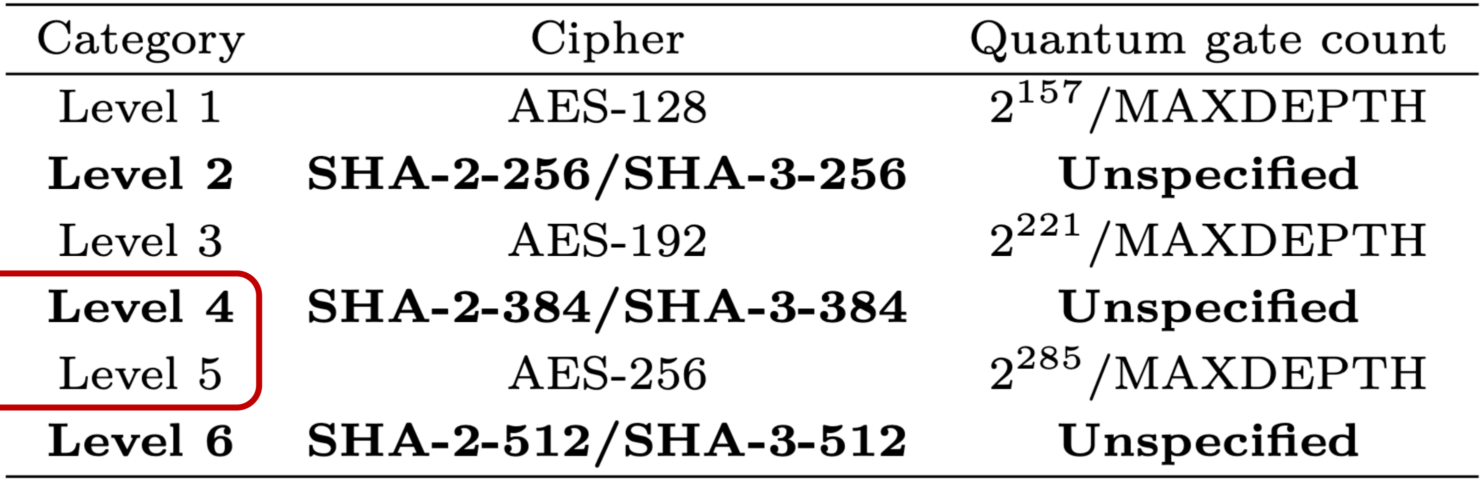
| Search Complexity | Category | Cipher | Quantum gate count |
|---|---|---|---|
| $2^{64}$ (key search) | Level 1 | AES-128 | $2^{157}$/MAXDEPTH |
| **$2^{85\sim}$ (BHT)** | **Level 2** | **SHA-2-256/SHA-3-256** | **Unspecified** |
| $2^{96}$ (key search) | Level 3 | AES-192 | $2^{221}$/MAXDEPTH |
| **$2^{128}$ (BHT)** | **Level 4** | **SHA-2-384/SHA-3-384** | **Unspecified** |
| $2^{128}$ (key search) | Level 5 | AES-256 | $2^{285}$/MAXDEPTH |
| **$2^{170\sim}$ (BHT)** | **Level 6** | **SHA-2-512/SHA-3-512** | **Unspecified** |



[1] Note that, barring some truly surprising technological development during the standardization process, NIST will assume that the five security strengths are correctly ordered in terms of practical security. (E.g., NIST will assume that a brute-force collision attack on SHA-256 will be technologically feasible before a brute-force key search attack on AES-192.)

# Grover on Hash Functions

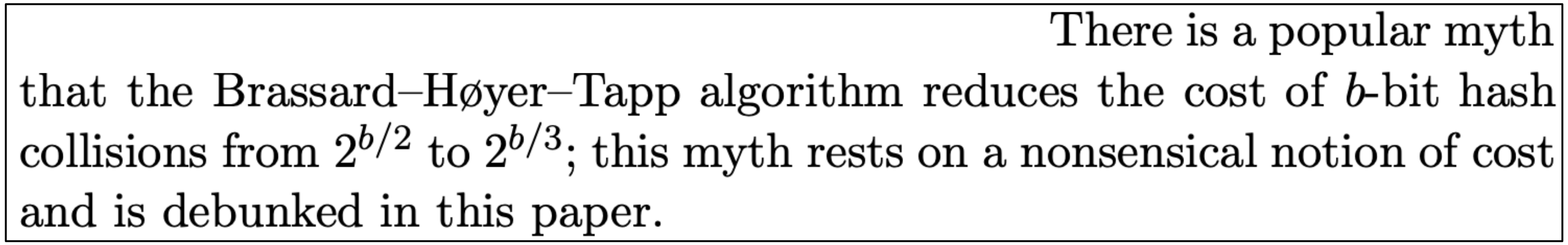
- **Levels 4, 5에 대한 Complexity (iteration)는 동일**
  → SHA2/3-384, AES 256에 대한 **양자 회로 비용에 따라 결정**됨



| Search Complexity | Category | Cipher | Quantum gate count |
|---|---|---|---|
| $2^{64}$ (key search) | Level 1 | AES-128 | $2^{157}$/MAXDEPTH |
| **$2^{85\sim}$ (BHT)** | **Level 2** | **SHA-2-256/SHA-3-256** | **Unspecified** |
| $2^{96}$ (key search) | Level 3 | AES-192 | $2^{221}$/MAXDEPTH |
| **$2^{128}$ (BHT)** | **Level 4** | **SHA-2-384/SHA-3-384** | **Unspecified** |
| $2^{128}$ (key search) | Level 5 | AES-256 | $2^{285}$/MAXDEPTH |
| **$2^{170\sim}$ (BHT)** | **Level 6** | **SHA-2-512/SHA-3-512** | **Unspecified** |

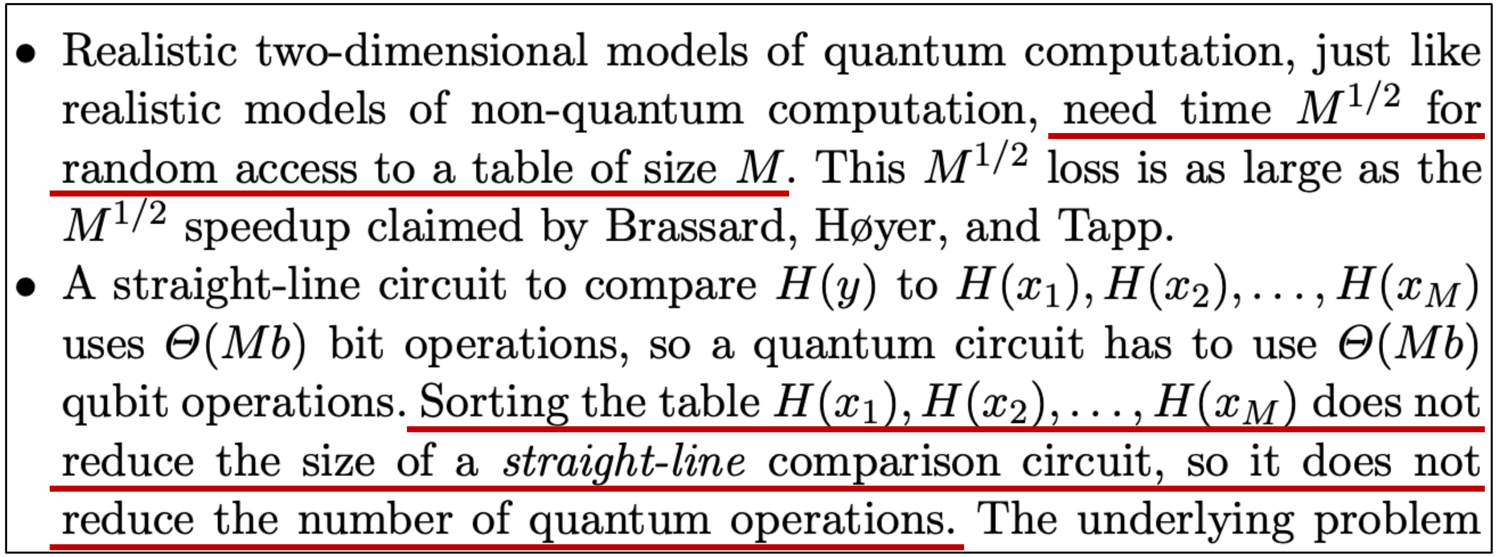
# Grover on Hash Functions

- **Consideration # 1**
- **BHT algorithm의 $O(2^{n/3})$ 는 이상적인 복잡도** (두 가지 이유)



There is a popular myth that the Brassard–Høyer–Tapp algorithm reduces the cost of $b$-bit hash collisions from $2^{b/2}$ to $2^{b/3}$; this myth rests on a nonsensical notion of cost and is debunked in this paper.

- **1.** Quantum ram access 및 size 비용
- 2. Search수는 Grover에 의해 줄어들지만, 내부의 해시 값 비교 step은 줄어들지 않음



- Realistic two-dimensional models of quantum computation, just like realistic models of non-quantum computation, need time $M^{1/2}$ for random access to a table of size $M$. This $M^{1/2}$ loss is as large as the $M^{1/2}$ speedup claimed by Brassard, Høyer, and Tapp.
- A straight-line circuit to compare $H(y)$ to $H(x_1), H(x_2), \ldots, H(x_M)$ uses $\Theta(Mb)$ bit operations, so a quantum circuit has to use $\Theta(Mb)$ qubit operations. Sorting the table $H(x_1), H(x_2), \ldots, H(x_M)$ does not reduce the size of a *straight-line* comparison circuit, so it does not reduce the number of quantum operations. The underlying problem

# Grover on Hash Functions

- Asiacrypt17에서, Chailloux, Naya-Plasencia, and Schrottenloher은 효율적인 양자 collision search 알고리즘을 제시 → **CNS 알고리즘**
  - **$O(2^{2n/5})$의 복잡도를 가지며, Quantum ram이 필요 없음**
    - **BHT 알고리즘 보다 높지만, 사실 BHT는 논란도 많으며 과도하게 비현실적**
  - CNS 알고리즘은 대신 $O(2^{n/5})$의 Classical 메모리가 필요함

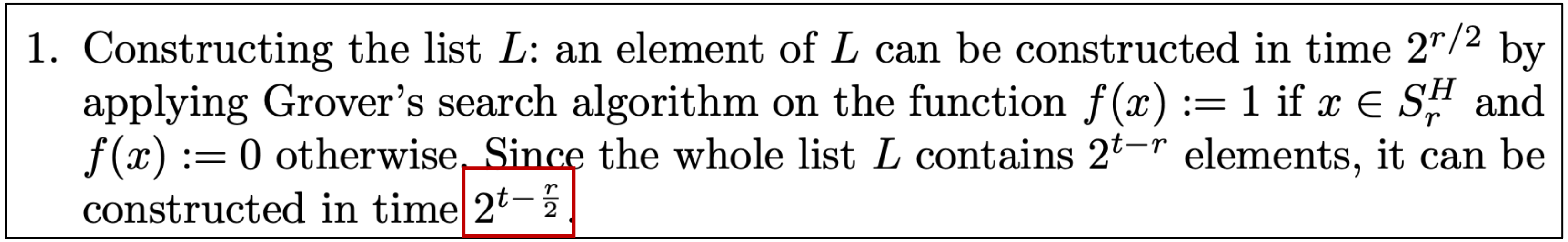
# Grover on Hash Functions

- 크게 2 가지 단계로 구성됨
  - **List 구성** 그리고 **Quantum Amplitude Amplification (QAA)** 알고리즘을 사용한 collision search

**Phase 1.**

**$S_r^H$ : 해시 함수의 output의 상위 r 개의 bit가 0으로 구성되는 input-output 쌍**

**$S_r^H$ 들로 구성되는 List $L$ 을 구성, List 크기는 $2^{t-r}$ → Grover를 사용하면 list를 채우는 complexity가 $2^{r/2}$ 로 감소**



1. Constructing the list $L$: an element of $L$ can be constructed in time $2^{r/2}$ by applying Grover's search algorithm on the function $f(x) := 1$ if $x \in S_r^H$ and $f(x) := 0$ otherwise. Since the whole list $L$ contains $2^{t-r}$ elements, it can be constructed in time $\boxed{2^{t-\frac{r}{2}}}$.

첫 Phase 1에 대한 복잡도

# Optimal

$t = 3n/5$

$r = 2n/5$

# Grover on Hash Functions

- 크게 2 가지 단계로 구성됨
  - **List 구성** 그리고 **Quantum Amplitude Amplification (QAA)** 알고리즘을 사용한 collision search

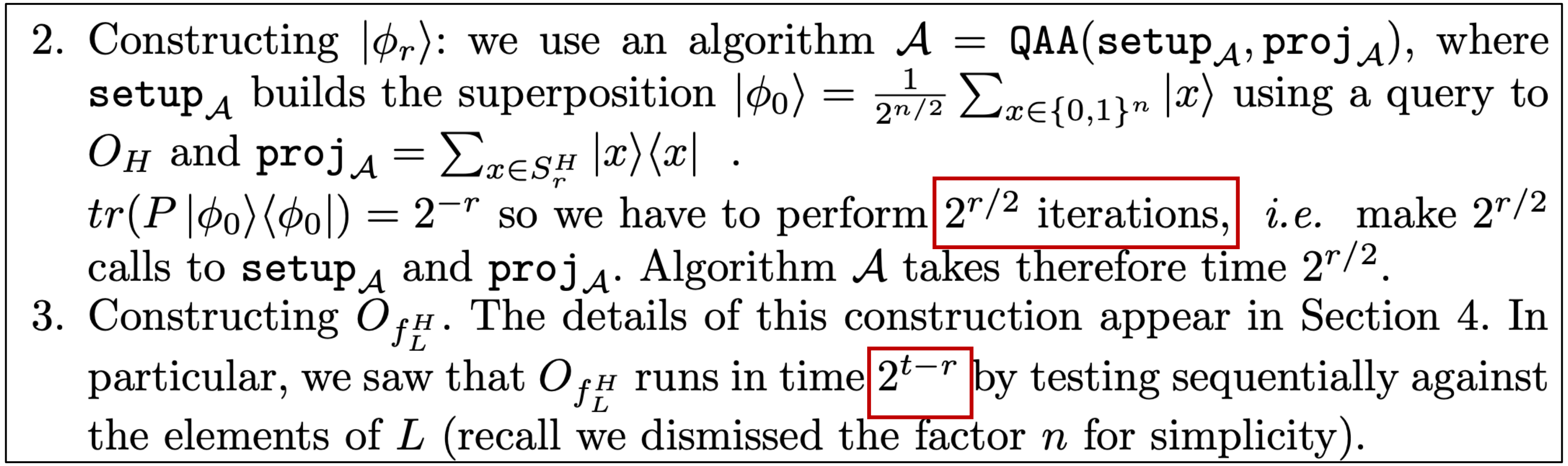
**Phase 2.**

**$S_r^H$ (x, H(x)) 에대한 $F_L^H$**

**(x’, H(x’)) ∈ L 이며, H(x) = H(x’) 을 찾는 경우 $F_L^H$=1을 반환 아닌 경우 0**

**1인 경우, (x , H(x) ∉ L 이지만 H(x) = H(x’) 의 확률이 높음 → Collision**

**→ 이걸 QAA로 찾음**

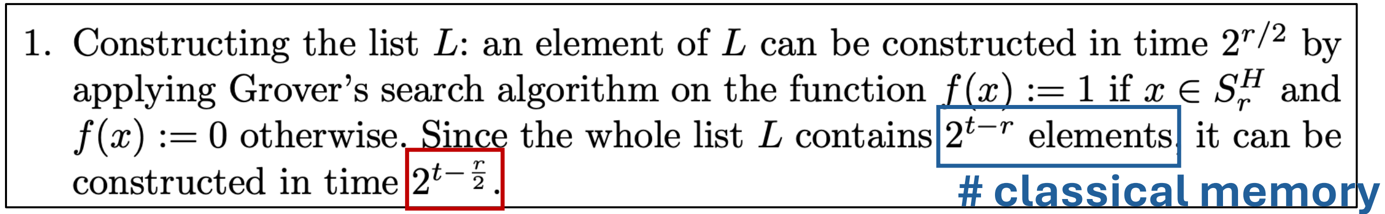


2. Constructing $|\phi_r\rangle$: we use an algorithm $\mathcal{A} = \texttt{QAA}(\texttt{setup}_{\mathcal{A}}, \texttt{proj}_{\mathcal{A}})$, where $\texttt{setup}_{\mathcal{A}}$ builds the superposition $|\phi_0\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle$ using a query to $O_H$ and $\texttt{proj}_{\mathcal{A}} = \sum_{x \in S_r^H} |x\rangle\langle x|$ . $tr(P\,|\phi_0\rangle\langle\phi_0|) = 2^{-r}$ so we have to perform $\boxed{2^{r/2}\text{ iterations,}}$ *i.e.* make $2^{r/2}$ calls to $\texttt{setup}_{\mathcal{A}}$ and $\texttt{proj}_{\mathcal{A}}$. Algorithm $\mathcal{A}$ takes therefore time $2^{r/2}$.
3. Constructing $O_{f_L^H}$. The details of this construction appear in Section 4. In particular, we saw that $O_{f_L^H}$ runs in time $\boxed{2^{t-r}}$ by testing sequentially against the elements of $L$ (recall we dismissed the factor $n$ for simplicity).

**2 + 3 을 $2^{\frac{n-t-1}{2}}$ 번 반복**

# Grover on Hash Functions

**Phase 1.** $S_r^H$ **들로 구성되는 List $L$ 을 구성, List 크기는 $2^{t-r}$ → Grover를 사용하면 list를 채우는 complexity가 $2^{r/2}$ 로 감**



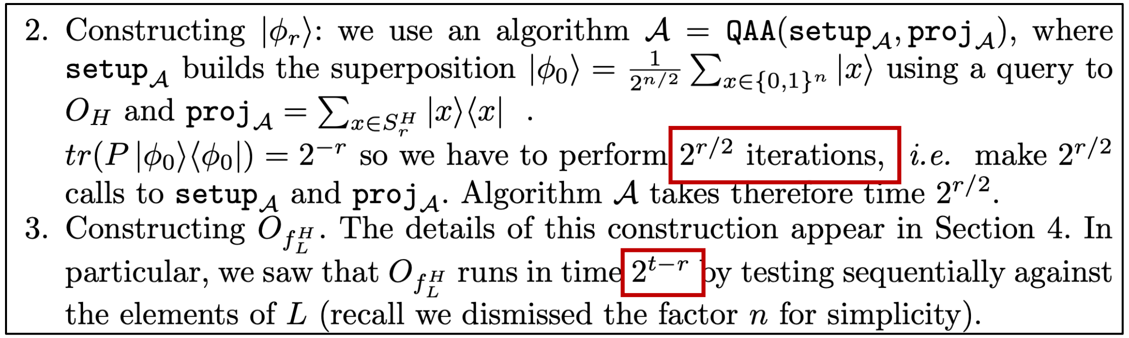
1. Constructing the list $L$: an element of $L$ can be constructed in time $2^{r/2}$ by applying Grover's search algorithm on the function $f(x) := 1$ if $x \in S_r^H$ and $f(x) := 0$ otherwise. Since the whole list $L$ contains $\boxed{2^{t-r} \text{ elements}}$, it can be constructed in time $\boxed{2^{t-\frac{r}{2}}}$.

**# classical memory**

**Phase 2.** $S_r^H$ **(x, H(x)) 에대한** $F_L^H$

**(x', H(x')) ∈ L 이며, H(x) = H(x') 을 찾는 경우 $F_L^H$=1을 반환 아닌 경우 0**
**1인 경우, (x , H(x) ∉ L 이지만 H(x) = H(x') 의 확률이 높음 → Collision**

**→ 이걸 QAA로 찾음**



2. Constructing $|\phi_r\rangle$: we use an algorithm $\mathcal{A} = \texttt{QAA}(\texttt{setup}_{\mathcal{A}}, \texttt{proj}_{\mathcal{A}})$, where $\texttt{setup}_{\mathcal{A}}$ builds the superposition $|\phi_0\rangle = \frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle$ using a query to $O_H$ and $\texttt{proj}_{\mathcal{A}} = \sum_{x\in S_r^H} |x\rangle\langle x|$ .
$tr(P\,|\phi_0\rangle\langle\phi_0|) = 2^{-r}$ so we have to perform $\boxed{2^{r/2} \text{ iterations,}}$ *i.e.* make $2^{r/2}$ calls to $\texttt{setup}_{\mathcal{A}}$ and $\texttt{proj}_{\mathcal{A}}$. Algorithm $\mathcal{A}$ takes therefore time $2^{r/2}$.
3. Constructing $O_{f_L^H}$. The details of this construction appear in Section 4. In particular, we saw that $O_{f_L^H}$ runs in time $\boxed{2^{t-r}}$ by testing sequentially against the elements of $L$ (recall we dismissed the factor $n$ for simplicity).

**2 + 3 을** $2^{\frac{n-t-1}{2}}$ **번 반복**

# Optimal
$t = 3n/5$
$r = 2n/5$

**Total:** $2^{(n-t-1)/2}(2^{r/2} + 2^{t-r}) + 2^{t-2/r}$ → $O(2^{2n/5})$

# Grover on Hash Functions

- NIST의 post-quantum security level을 고려했을 때, **CNS 알고리즘 복잡도**
  - **여전히 적절한 기준선을 제공하기 어려움**

Search Complexity

$2^{64}$ (key search)

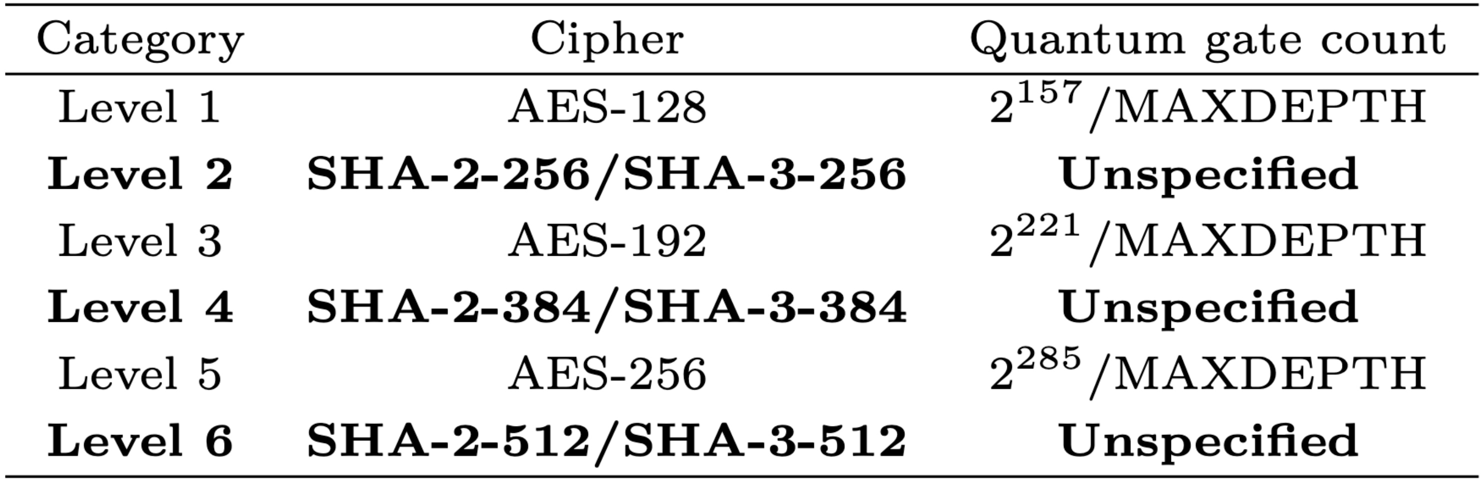
**$2^{102\sim}$ (CNS)**

$2^{96}$ (key search)

**$2^{153\sim}$ (CNST)**

$2^{128}$ (key search)

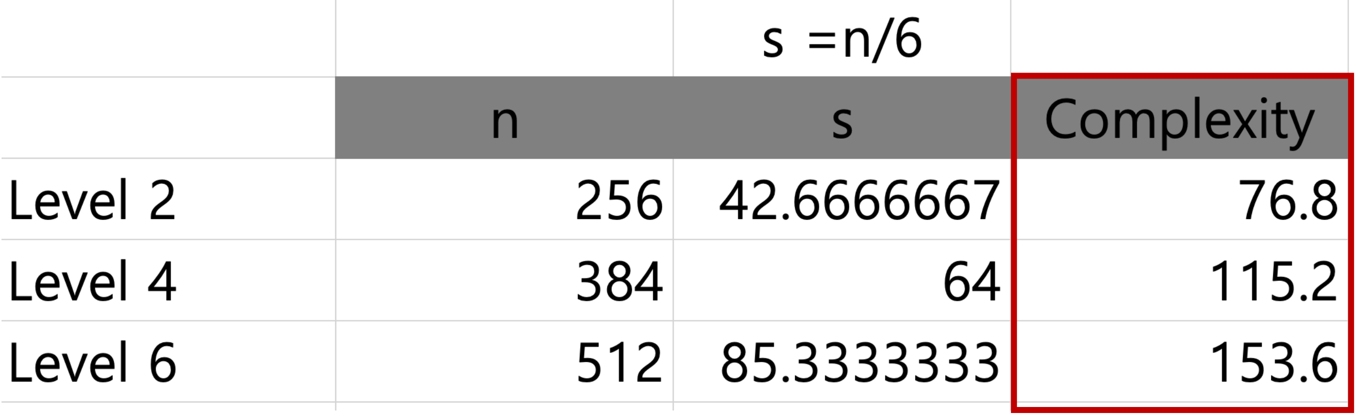
**$2^{204\sim}$ (CNS)**



| Category | Cipher | Quantum gate count |
|---|---|---|
| Level 1 | AES-128 | $2^{157}$/MAXDEPTH |
| **Level 2** | **SHA-2-256/SHA-3-256** | **Unspecified** |
| Level 3 | AES-192 | $2^{221}$/MAXDEPTH |
| **Level 4** | **SHA-2-384/SHA-3-384** | **Unspecified** |
| Level 5 | AES-256 | $2^{285}$/MAXDEPTH |
| **Level 6** | **SHA-2-512/SHA-3-512** | **Unspecified** |

# Grover on Hash Functions

- 하지만 CNS 알고리즘을 병렬화 할 경우, 복잡도를 감소시킬 수 있음
  - **2^s 개의 Quantum machine을 병렬로 운용할 경우 복잡도는 다음과 같이 감소함**
    - $\boldsymbol{O\left(2^{2n/5} - 2^{3s/5}\right)}$
  - 최대 병렬화: $s \leq n/4 \quad (n = 256, 384, 512)$
- Levels 2, 4, 6에 대한 적정 복잡도를 제공하기 위해 $\boldsymbol{s = n/6}$ **으로 설정**

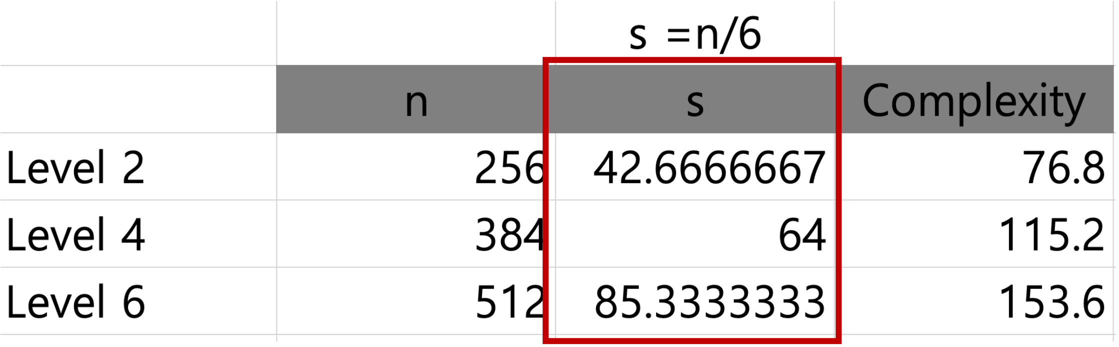


| | | s =n/6 | |
|---|---|---|---|
| | n | s | Complexity |
| Level 2 | 256 | 42.6666667 | 76.8 |
| Level 4 | 384 | 64 | 115.2 |
| Level 6 | 512 | 85.3333333 | 153.6 |

# Grover on Hash Functions

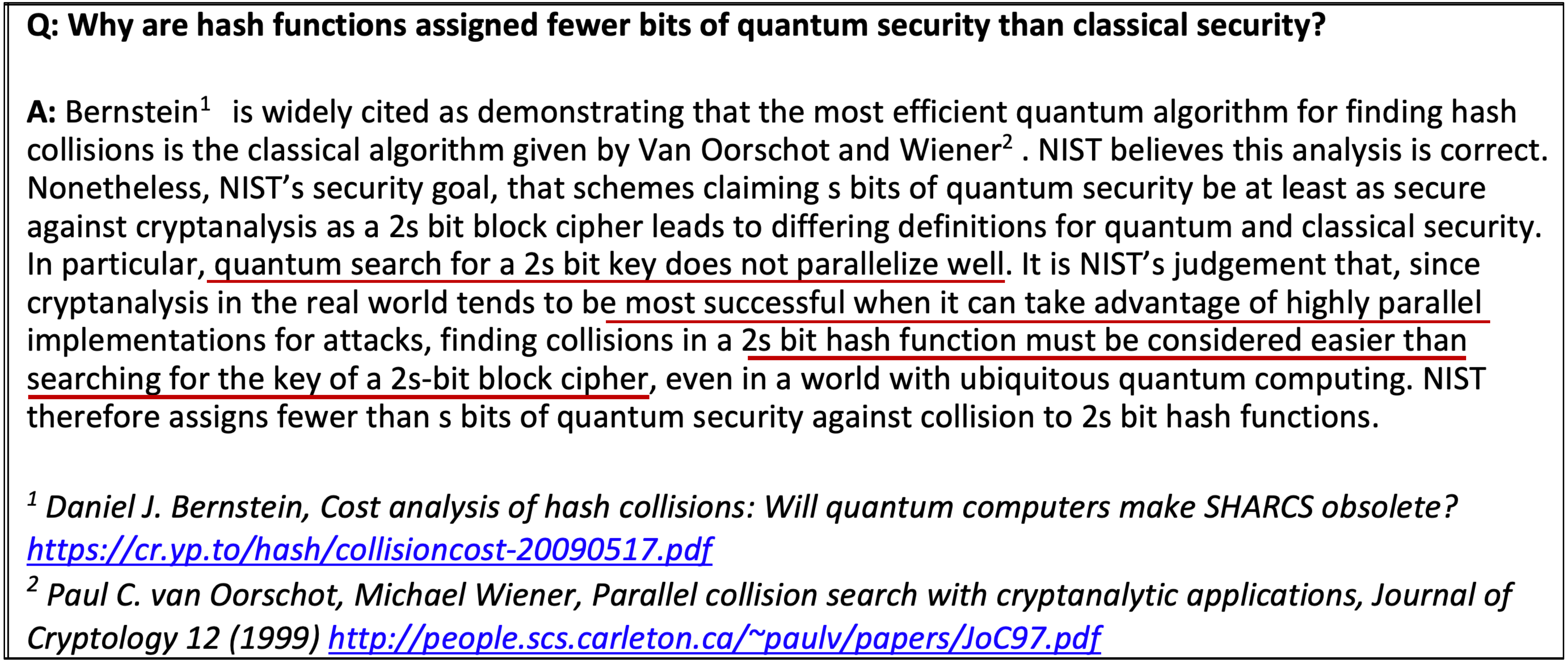
- 병렬화를 가정한 CNS 알고리즘 복잡도
  - 적절한 기준을 제공할 수 있음, 하지만 큐빗 수가 엄청 남 (그래도 이대로 진행)

| Search Complexity | Category | Cipher | Quantum gate count |
|---|---|---|---|
| $2^{64}$ (key search) | Level 1 | AES-128 | $2^{157}$/MAXDEPTH |
| **$2^{76\sim}$ (CNS)** | **Level 2** | **SHA-2-256/SHA-3-256** | **Unspecified** |
| $2^{96}$ (key search) | Level 3 | AES-192 | $2^{221}$/MAXDEPTH |
| **$2^{115\sim}$ (CNST)** | **Level 4** | **SHA-2-384/SHA-3-384** | **Unspecified** |
| $2^{128}$ (key search) | Level 5 | AES-256 | $2^{285}$/MAXDEPTH |
| **$2^{153\sim}$ (CNS)** | **Level 6** | **SHA-2-512/SHA-3-512** | **Unspecified** |



| | | s =n/6 | |
|---|---|---|---|
| | n | s | Complexity |
| Level 2 | 256 | 42.6666667 | 76.8 |
| Level 4 | 384 | 64 | 115.2 |
| Level 6 | 512 | 85.3333333 | 153.6 |

# Grover on Hash Functions



**Q: Why are hash functions assigned fewer bits of quantum security than classical security?**

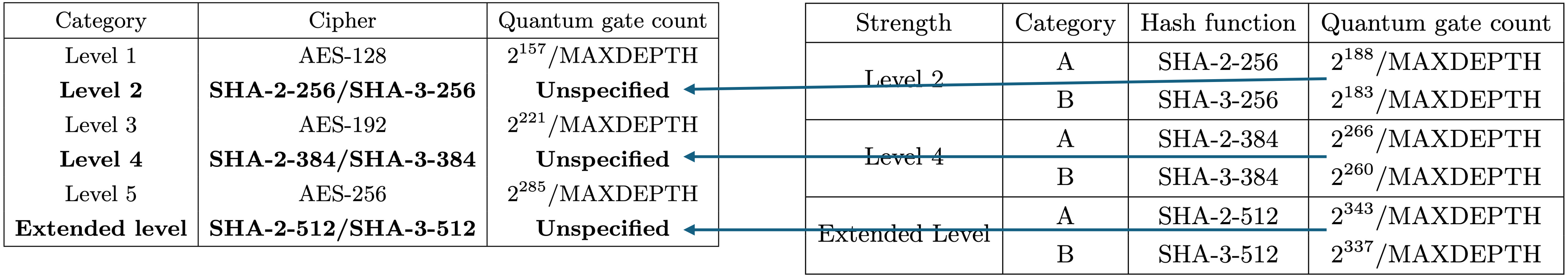
**A:** Bernstein[1] is widely cited as demonstrating that the most efficient quantum algorithm for finding hash collisions is the classical algorithm given by Van Oorschot and Wiener[2] . NIST believes this analysis is correct. Nonetheless, NIST's security goal, that schemes claiming s bits of quantum security be at least as secure against cryptanalysis as a 2s bit block cipher leads to differing definitions for quantum and classical security. In particular, quantum search for a 2s bit key does not parallelize well. It is NIST's judgement that, since cryptanalysis in the real world tends to be most successful when it can take advantage of highly parallel implementations for attacks, finding collisions in a 2s bit hash function must be considered easier than searching for the key of a 2s-bit block cipher, even in a world with ubiquitous quantum computing. NIST therefore assigns fewer than s bits of quantum security against collision to 2s bit hash functions.

[1] *Daniel J. Bernstein, Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?* *https://cr.yp.to/hash/collisioncost-20090517.pdf*

[2] *Paul C. van Oorschot, Michael Wiener, Parallel collision search with cryptanalytic applications, Journal of Cryptology 12 (1999)* *http://people.scs.carleton.ca/~paulv/papers/JoC97.pdf*

# Grover on Hash Functions

- BHT algorithm에서 CNS 병렬화로 변경
  - **적절 기준선 제공 가능, 문제는 큐빗**

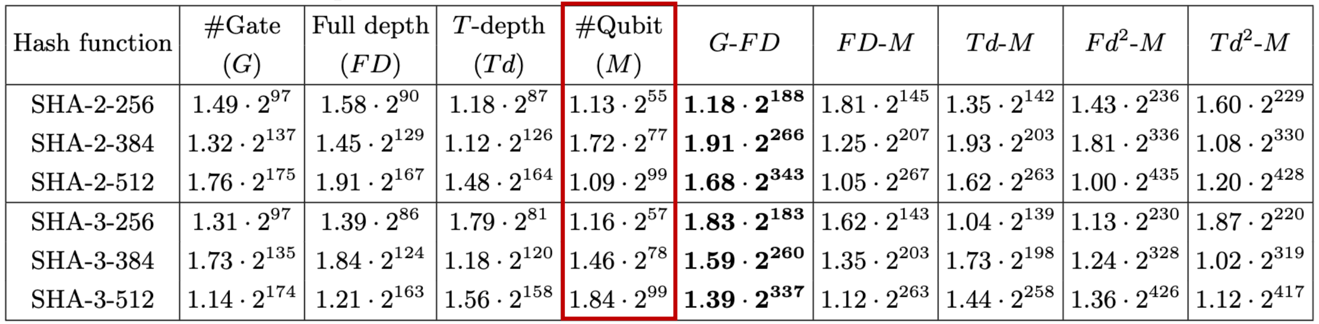


| Category | Cipher | Quantum gate count |
|---|---|---|
| Level 1 | AES-128 | $2^{157}$/MAXDEPTH |
| **Level 2** | **SHA-2-256/SHA-3-256** | **Unspecified** |
| Level 3 | AES-192 | $2^{221}$/MAXDEPTH |
| **Level 4** | **SHA-2-384/SHA-3-384** | **Unspecified** |
| Level 5 | AES-256 | $2^{285}$/MAXDEPTH |
| **Extended level** | **SHA-2-512/SHA-3-512** | **Unspecified** |

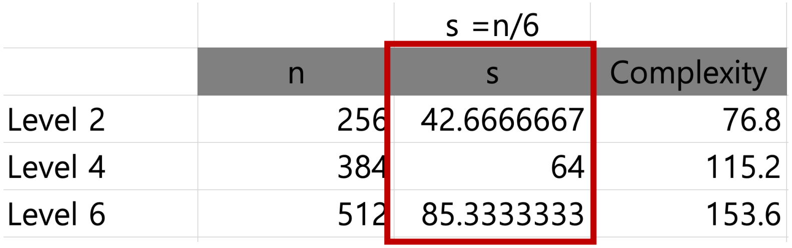
Table 2: Security levels defined in this work.

| Strength | Category | Hash function | Quantum gate count |
|---|---|---|---|
| Level 2 | A | SHA-2-256 | $2^{188}$/MAXDEPTH |
| | B | SHA-3-256 | $2^{183}$/MAXDEPTH |
| Level 4 | A | SHA-2-384 | $2^{266}$/MAXDEPTH |
| | B | SHA-3-384 | $2^{260}$/MAXDEPTH |
| Extended Level | A | SHA-2-512 | $2^{343}$/MAXDEPTH |
| | B | SHA-3-512 | $2^{337}$/MAXDEPTH |

Table 13: AND-based quantum resources for quantum collision search on SHA-2 and SHA-3.



| Hash function | #Gate ($G$) | Full depth ($FD$) | $T$-depth ($Td$) | #Qubit ($M$) | $G$-$FD$ | $FD$-$M$ | $Td$-$M$ | $Fd^2$-$M$ | $Td^2$-$M$ |
|---|---|---|---|---|---|---|---|---|---|
| SHA-2-256 | $1.49 \cdot 2^{97}$ | $1.58 \cdot 2^{90}$ | $1.18 \cdot 2^{87}$ | $1.13 \cdot 2^{55}$ | $\mathbf{1.18 \cdot 2^{188}}$ | $1.81 \cdot 2^{145}$ | $1.35 \cdot 2^{142}$ | $1.43 \cdot 2^{236}$ | $1.60 \cdot 2^{229}$ |
| SHA-2-384 | $1.32 \cdot 2^{137}$ | $1.45 \cdot 2^{129}$ | $1.12 \cdot 2^{126}$ | $1.72 \cdot 2^{77}$ | $\mathbf{1.91 \cdot 2^{266}}$ | $1.25 \cdot 2^{207}$ | $1.93 \cdot 2^{203}$ | $1.81 \cdot 2^{336}$ | $1.08 \cdot 2^{330}$ |
| SHA-2-512 | $1.76 \cdot 2^{175}$ | $1.91 \cdot 2^{167}$ | $1.48 \cdot 2^{164}$ | $1.09 \cdot 2^{99}$ | $\mathbf{1.68 \cdot 2^{343}}$ | $1.05 \cdot 2^{267}$ | $1.62 \cdot 2^{263}$ | $1.00 \cdot 2^{435}$ | $1.20 \cdot 2^{428}$ |
| SHA-3-256 | $1.31 \cdot 2^{97}$ | $1.39 \cdot 2^{86}$ | $1.79 \cdot 2^{81}$ | $1.16 \cdot 2^{57}$ | $\mathbf{1.83 \cdot 2^{183}}$ | $1.62 \cdot 2^{143}$ | $1.04 \cdot 2^{139}$ | $1.13 \cdot 2^{230}$ | $1.87 \cdot 2^{220}$ |
| SHA-3-384 | $1.73 \cdot 2^{135}$ | $1.84 \cdot 2^{124}$ | $1.18 \cdot 2^{120}$ | $1.46 \cdot 2^{78}$ | $\mathbf{1.59 \cdot 2^{260}}$ | $1.35 \cdot 2^{203}$ | $1.73 \cdot 2^{198}$ | $1.24 \cdot 2^{328}$ | $1.02 \cdot 2^{319}$ |
| SHA-3-512 | $1.14 \cdot 2^{174}$ | $1.21 \cdot 2^{163}$ | $1.56 \cdot 2^{158}$ | $1.84 \cdot 2^{99}$ | $\mathbf{1.39 \cdot 2^{337}}$ | $1.12 \cdot 2^{263}$ | $1.44 \cdot 2^{258}$ | $1.36 \cdot 2^{426}$ | $1.12 \cdot 2^{417}$ |



| | | s =n/6 | |
|---|---|---|---|
| | n | s | Complexity |
| Level 2 | 256 | 42.6666667 | 76.8 |
| Level 4 | 384 | 64 | 115.2 |
| Level 6 | 512 | 85.3333333 | 153.6 |

# Thank you!