Quantum Collision Attacks on Reduced SHA-256 and SHA-512

Akinori Hosoyamada[1,2] and Yu Sasaki[1]

[1] NTT Secure Platform Laboratories, Tokyo, Japan,
{akinori.hosoyamada.bh,yu.sasaki.sk}@hco.ntt.co.jp
[2] Nagoya University, Nagoya, Japan, hosoyamada.akinori@nagoya-u.jp

Abstract. In this paper, we study dedicated quantum collision attacks on SHA-256 and SHA-512 for the first time. The attacks reach 38 and 39 steps, respectively, which significantly improve the classical attacks for 31 and 27 steps. Both attacks adopt the framework of the previous work that converts many semi-free-start collisions into a 2-block collision, and are faster than the generic attack in the cost metric of time-space tradeoff. We observe that the number of required semi-free-start collisions can be reduced in the quantum setting, which allows us to convert the previous classical 38 and 39 step semi-free-start collisions into a collision. The idea behind our attacks is simple and will also be applicable to other cryptographic hash functions.

Keywords: symmetric key cryptography, hash function, SHA-256, SHA-512, collision attack, quantum attack, conversion from semi-free-start collisions

1 Introduction

Cryptographic hash functions take an arbitrary length message as input and generate a fixed-length bit string. One of the most important security criteria is collision resistance. For a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$, the complexity to find two distinct values x_1 and x_2 such that $\mathcal{H}(x_1) = \mathcal{H}(x_2)$ should be $O(2^{n/2})$. The collision resistance is a practically relevant notion. For example, Stevens et al. [42], in their attack against SHA-1, forged two PDF documents with the same hash digest that display different arbitrarily-chosen visual contents.

The SHA-2 family is one of the most important hash functions at the present time, which is specified and standardized by NIST [37]. There are two core algorithms; SHA-256 and SHA-512, depending on the word size. Moreover four schemes are additionally specified depending on the output size. SHA-2 are used in wide range of communication protocols such as TLS/SSL, SSH, and IPsec. SHA-2 are also used by the digital currency such as Bitcoin. After the recent break of SHA-1 [24], industry accelerated the migration to SHA-2.

History of SHA-2 Cryptanalysis. SHA-2 received a massive amount of security analysis. Preimeage attacks were studied in [20,3,16,22] and a conversion to

pseudo-collisions was studied in [25]. Those would work relatively a large number of rounds, say 52 out of 64 steps of SHA-256 [22], while those only achieve a marginal amount of speed up. Those are interesting theoretical results but not strongly related to this research. As a non-random property, second-order differential collisions defined over four distinct inputs were studied [5].

More relevant works to this research are the attempts to apply previous collision finding techniques to SHA-2 or to find collisions on reduced-step SHA-2. The challenge to find a collision on reduced-step SHA-2 was initiated by [34], which found a collision on 19 out of 64 steps of SHA-224. This is a pioneering work to construct differential characteristic only having a single local collision. Then, this type of local collisions were manually optimized to find collisions of 21 steps of SHA-256 [38], and later improved to 22 steps [40], and to 24 steps and extended to SHA-512 [19,41]. However, it was indicated that the local collision by [38] could work only up to 24 rounds [19] and indeed this was the last work for improving the manually detected local collision.

The most recent technical innovation is the development of automated differential characteristic search tools, which was initiated by Mendel et al. [30] to find a collision on 27 steps of SHA-256. Because of the search space, the efficiency of the algorithm is crucial for the automated search tool. Mendel et al. improved the algorithm and presented a 31-step collision attack and a 38-step semi-free-start collision attack against SHA-256 [32]. [3] This is the current best (semi-free-start) collision attacks for SHA-256. The algorithm was further improved to apply it to SHA-512 [13], SHA-512/224 and SHA-512-256 [10]. For SHA-512, 27-step collisions and 39-step semi-free-start collisions [10] are the current best results.

Techniques for Finding SHA-2 Collisions. For the attack on SHA-256, Mendel et al. [32] presented a framework to convert semi-free-start collision attacks having some special property into a 2-block collision. The framework is illustrated in Fig. 1. The attacker first analyzes the second block without fixing IV for the second block, $\mathrm{IV}_{\mathrm{second}}$. A semi-free-start collision attack that can work for 2^X choices, typically for any unfixed X bits, of $\mathrm{IV}_{\mathrm{second}}$ is located in the second block. Then, the attacker tests 2^{n-X} messages for the first block to hit one of 2^X choices of $\mathrm{IV}_{\mathrm{second}}$, typically to hit the fixed $n-X$ bits of $\mathrm{IV}_{\mathrm{second}}$. Finally, the attacker determines the rest part of the second block to generate a 2-block collision.

The cost for the first block is 2^{256-X} for SHA-256. To be faster than the birthday paradox, X must satisfy $X > 128$. To achieve such a semi-free-start collision attack, the previous work [32] generated a differential characteristic such that the characteristic can be satisfied for any value of the first five message words. Hence, it achieves $X = 160$. (As explained later, those five message words can be adjusted to achieve a fixed 160-bit internal state value for any 160 bits of $\mathrm{IV}_{\mathrm{second}}$.)

[3] For readers who are not familiar with various types of collisions, we explain the difference among collisions, semi-free-stard collisions, and free-start collisions in Section A.

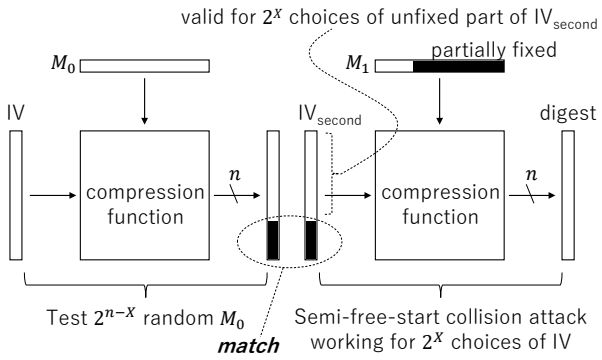

Fig. 1. Converting Semi-free-start Collisions into 2-block Collisions.

Dedicated Quantum Collision Attacks. Recently, it has been shown that collision attacks on hash functions with quantum machines can break more rounds than the attacks with classical machines [17]. Whether a hash function is attacked or not is judged by comparing the complexity of the generic attack (birthday paradox) and a dedicated attack. To find a collision, dedicated attacks mostly apply differential cryptanalysis. With quantum machines, the speed of finding a value satisfying a differential characteristic becomes a square root compared to classical machines, while the speed of the generic collision attack cannot be a square root of the birthday paradox, $O(2^{n/4})$. Indeed, the tight bound of the query complexity to find a collision was proven to be $O(2^{n/3})$ [44]. As a result, dedicated attacks can be stronger when quantum machines are available. In fact, such cases were observed for AES hashing modes [17,12] and Gimli [14].

In the quantum setting, the generic attack complexity of finding collisions depends on settings about the resource that an attacker can use. The previous work discussed three settings. In the first setting, a small (polynomial size) quantum computer and a large (exponential size) qRAM. In the second setting, a small (polynomial size) quantum computer and a large (exponential size) classical memory, In the third setting, efficiency of quantum algorithms are evaluated by their time-space tradeoff.

In this paper, we focus on the third setting, of which details are as follows. Note that we do not take error corrections into account and consider that the running time of a quantum circuit is proportional to the depth of the circuit.

Cost metric of time-space tradeoff. The efficiency of an attack is evaluated by the tradeoff between T and S, where T is the attack time complexity (or, the depth of the quantum circuit) and S is the hardware size required for the attack (i.e., S is the maximum size of quantum computers (or, width of quantum circuits) and classical computers). S can be exponentially large, and we do not make distinction between qubits for computation and qubits for memory. Bernstein [4] observed that, when a *classical* computer of size S

is available, by using the parallel rho method [39] we can find a collision of a random function in time $T = O(2^{n/2}/S)$. There does not exist a quantum attack on a random function that achieves a better tradeoff than this classical attack.[4] Hence, a dedicated quantum collision attack on a concrete hash function that uses a quantum computer of size S is considered to be valid if its time complexity T is less than $2^{n/2}/S$.

The condition $T < 2^{n/2}/S$ is equivalent to $T \cdot S < 2^{n/2}$. Hence the efficiency of a quantum attack in the time-space tradeoff metric is evaluated by the multiplication of T and S, and the threshold for the attack to be valid is $2^{n/2}$.

Jaques and Schanck [21] showed that when error correction is necessary and quantum memory is actively corrected, it is realistic to model that the cost of a quantum attack is proportional to the multiplication of the depth and the width of the quantum circuit used in the attack. Therefore, although we do not care about error corrections in our complexity analysis, the cost metric of time-space tradeoff is in fact reasonable from the view point of cost estimation *including* quantum error correction (when quantum memory is actively corrected).

Research Challenge. The collision resistance of SHA-2 family in the quantum setting has not been studied before. [5] In fact, this is not a simple task. As mentioned before, the current differential characteristics for SHA-2 collision attacks consist of a single local collision. The previous work showed [17] that the cost to satisfy an uncontrolled part of the differential characteristic can be square root, while the differential characteristic for SHA-2 does not have such a form. Thus this issue deserves careful investigation.

Our Contributions. In this paper, we present quantum collision attacks on SHA-256 and on SHA-512 that break more rounds than the attacks in the classical setting. Our attacks are valid in the time-space tradeoff cost metric. The number of attacked steps is compared in Table 1.

To generate collisions, we follow the same approach as the previous work. Namely, we locate a semi-free-start collision in the second block and find a first-block message to hit one of IVs that is acceptable for the second block. In the previous work, it is principally inevitable that the semi-free-start collision attack must work for at least 2^X choices of IVs, where $X > 128$ for SHA-256. This is a strong requirement, which significantly restricts the search space to find a suitable differential characteristic. We observe that if quantum machines are available, we can construct an attack with an intuitive condition of $X > 0$ by ignoring the constant factor. In practice, the constant factor cannot be ignored and we will show a rigorous complexity analysis.

[4] There is no proof that the bound $O(2^{n/2}/S)$ is the best, but achieving a better bound is hard.

[5] From the view point of provable security, there is a previous work that suggests that the SHA-2 mode is reasonable in the quantum setting [18].

Table 1. Comparison of the Attack Results. The quantum attacks on SHA-256 and SHA-512 are faster than the generic attack as long as $S < 2^{12}$ and $S < 2^{6.6}$, respectively.

Target	Setting	Type	Steps	Complexity	Reference
SHA-256	classic	collision	28/64	practical	[32]
	classic	collision	31/64	$2^{65.5}$	[32]
	classic	semi-free-start collision	38/64	practical	[32]
	quantum	collision	38/64	$2^{122}/\sqrt{S}$	Section 5
SHA-512	classic	collision	24/80	practical	[19,41]
	classic	collision	27/80	practical	[10]
	classic	semi-free-start collision	38/80	practical	[13]
	classic	semi-free-start collision	39/80	practical	[10]
	quantum	collision	39/80	$2^{252.7}/\sqrt{S}$	Section 6

For SHA-256, the previous work [32] found a differential characteristic with $X > 128$ up to 31 steps, while unconditioned semi-free-start collisions could be generated for 38 steps. Hence we start from the 38-step semi-free-start collision example generated by [32] and slightly modify its message words so that semi-free-start collisions can be generated for multiple IVs. We achieve $X \approx 20$ for 38-step SHA-256. If we have a quantum computer of size S, the attack complexity is about $c \cdot \sqrt{2^{256-20}/S} = 2^{122}/\sqrt{S}$, where c is a small constant and rigorous analysis shows $c \approx 2^4$. Because the generic attack cost under the time-space metric is $2^{128}/S$, our attack is faster than the generic attack when $S < 2^{12}$.

For SHA-512, it seems difficult to build a differential characteristic with a lot of degrees of freedom such as $X > 256$. In fact, the previous work [10] could not apply the 2-block conversion, and the current strategy is limited to be a single-block attack. In this paper, we observe that the 39-step semi-free-start collision attack [10] can accept multiple choices of IV with some X that is much smaller than 256, and will convert it into 2-block collision in the quantum setting.

As we mentioned before, the previous work [17] discussed three settings depending on available computational resources. In fact our attacks are valid only in the setting of time-space tradeoff because the time complexity exceed the generic complexity in other settings. Nevertheless, we would like to remark that dedicated attacks that are valid in this setting (including our attacks) are always better than the generic attacks in other settings from the viewpoint of time-space tradeoff. This is because the generic attacks in other settings have time-space tradeoff $T^2 \cdot S = 2^n$, which is worse than the trade-off $T \cdot S = 2^n$ of the generic attack in our setting. [6]

Some readers may think that our attacks are invalid because the margin of our attacks (compared to the generic attack) are too small while we do not take

[6] The generic attacks in other two settings are the BHT algorithm [7] and the CNS algorithm [8]. The BHT algorithm runs in time $T = O(2^{n/3})$ and uses $S = O(2^{n/3})$ qRAM. The CNS algorithm runs in time $T = O(2^{2n/5})$ and uses no qRAM, but requires $S = O(2^{n/5})$ classical memory.

the overhead for quantum computation, or their complexity does not significantly outperform the classical complexity. However, security of symmetric-key primitives is generally measured under the most vulnerable environment (they must resist any attacks in any nitpicked setting like $S = 1$). The principle of security under the most vulnerable environment makes it natural to ignore the overhead because the overhead for quantum computation may drastically be reduced by future technical developments. In addition, when reduced-step variants of symmetric-key primitives are analyzed, the most important factors is the number of attacked steps rather than the attack cost. Our quantum attacks break significantly more steps than the classical attacks.

Remark 1. For reference, we also provide discussions on comparison between our attacks and a generic collision attack based on the multi-target preimage search. See Section B for details.

Future Directions. Due to its simplicity, we believe that the idea of our 2-block quantum collision attacks is applicable to other cryptographic hash functions. It will also be interesting to study optimizations of differential characteristics for the classical semi-free-start collision attack with respect to the conversion to the quantum collision attack. Some observations and initial work will be provided in the last part of the paper.

Paper Organization. Section 2 is preliminaries. Section 3 explains the previous collision and semi-free-start collision attacks. Section 4 explains our observation that is used in our quantum attacks. Sections 5 and 6 show the attack algorithms and their evaluations. Section 7 provides discussion toward future applications of our attack idea. Finally, we conclude this paper in Section 8.

2 Preliminaries

For n-bit strings x and y, $\neg x$, $x \wedge y$, $x \vee y$ and $x \oplus y$ denote the bit-wise negation of x, the bit-wise AND on x and y, the bit-wise OR on x and y, and the bit-wise XOR on x and y, respectively. For an n-bit string x and a non-negative integer m such that $m \leq n$, $x \gg m$ (resp., $x \ggg m$) denotes the m-bit right shift operation on x (resp., the m-bit *circular* right shift operation on x). We identify the set of n-bit strings $\{0,1\}^n$ with the sets $\{0,\ldots,2^n-1\}$ and $\mathbb{Z}/2^n\mathbb{Z}$. $x+y$ denotes the modular addition of x and y for $x, y \in \mathbb{Z}/2^n\mathbb{Z}$, unless otherwise noted. Sometimes we use the symbol $\boxplus$ instead of $+$. We assume that readers are familiar with basics on quantum computation[7].

[7] Knowledge on quantum computations is required to fully understand our complexity analysis, though, essentially the quantum algorithms we use are only the (parallelized) Grover search, and we use them in an almost black-box manner.

2.1 Specification of SHA-256 and SHA-512

SHA-256 and SHA-512 adopt the Merkle-Damgård construction, and their compression functions adopt the Davies-Meyer construction. Let w be the word size, which is 32 for SHA-256 and 64 for SHA-512. The length of message blocks is $16w$ bits (512 bits for SHA-256 and 1024 bits for SHA-512), and the length of chaining values and final outputs is $8w$ bits (256 bits for SHA-256 and 512 bits for SHA-512).

Given a chaining value (or the initial value IV) $H = (H_0, \ldots, H_7) \in (\{0,1\}^w)^8$ and a message block $M = (M_0, \ldots, M_{15}) \in (\{0,1\}^w)^{16}$, the output value of the compression function $f(H, M)$ is computed by iteratively updating internal states as follows. The number of steps, which is denoted by r, is 64 for SHA-256 and 80 for SHA-512.

1. (Message expansion.) Compute W_i $(i = 0, \ldots, r-1)$ by

$$W_i := \begin{cases} M_i & \text{for } i = 0, \ldots, 15, \\ \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16} & \text{for } i = 16, \ldots, r-1. \end{cases}$$

   The functions $\sigma_0, \sigma_1 : \{0,1\}^w \to \{0,1\}^w$ are defined later.
2. (Iterative state updates.) Set $st_{-1} := H$. For $i = 0, \ldots, r-1$, update the $8w$-bit state $st_{i-1} = (A_{i-1}, A_{i-2}, A_{i-3}, A_{i-4}, E_{i-1}, E_{i-2}, E_{i-3}, E_{i-4})$ to $st_i = (A_i, A_{i-1}, A_{i-2}, A_{i-3}, E_i, E_{i-1}, E_{i-2}, E_{i-3})$, where

$$\begin{aligned} E_i &:= E_{i-4} + A_{i-4} + \Sigma_1(E_{i-1}) + \mathrm{IF}(E_{i-1}, E_{i-2}, E_{i-3}) + K_i + W_i, \\ A_i &:= \Sigma_0(A_{i-1}) + \mathrm{MAJ}(A_{i-1}, A_{i-2}, A_{i-3}) + E_i - A_{i-4}. \end{aligned}$$

   The functions $\mathrm{IF}, \mathrm{MAJ} : (\{0,1\}^w)^3 \to \{0,1\}^w$ and $\Sigma_0, \Sigma_1 : \{0,1\}^w \to \{0,1\}^w$ are defined later. K_i is a step-dependent constant. Since the value of K_i does not affect our attacks, we omit the value of K_i. See also Fig. 2.
3. Compute the next chaining value $f(H, M)$ as $f(H, M) := st_{r-1} + H$. (Only here, the symbol "+" denotes the word-wise modular addition.)

The functions $\mathrm{IF}, \mathrm{MAJ} : (\{0,1\}^w)^3 \to \{0,1\}^w$ are defined as

$$\mathrm{IF}(x, y, z) = (x \wedge y) \oplus ((\neg x) \wedge z), \quad \mathrm{MAJ}(x, y, z) = (x \wedge y) \oplus (y \wedge z) \oplus (z \wedge x)$$

for both of SHA-256 and SHA-512. In addition, $\Sigma_0, \Sigma_1, \sigma_0, \sigma_1$ are defined by

$$\begin{aligned} \Sigma_0(x) &= (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22), \\ \sigma_0(x) &= (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3), \\ \Sigma_1(x) &= (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25), \\ \sigma_1(x) &= (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10) \end{aligned}$$

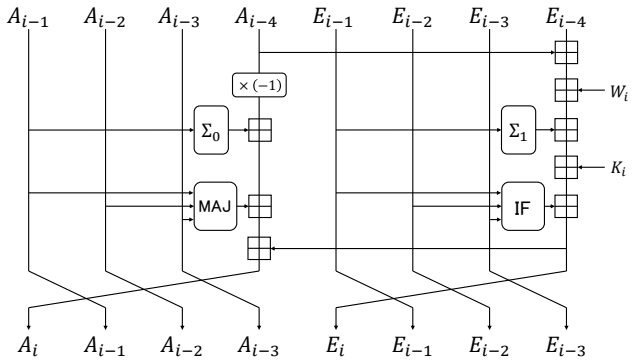

Fig. 2. This is an alternative representation of the state update function devised by the previous work [30]. The operation "$\times(-1)$" denotes the multiplication by (-1) in $\mathbb{Z}/2^w\mathbb{Z}$.

for SHA-256, and

$$\begin{aligned}
\Sigma_0(x) &= (x \ggg 28) \oplus (x \ggg 34) \oplus (x \ggg 39),\\
\sigma_0(x) &= (x \ggg 1) \oplus (x \ggg 8) \oplus (x \gg 7),\\
\Sigma_1(x) &= (x \ggg 14) \oplus (x \ggg 18) \oplus (x \ggg 41),\\
\sigma_1(x) &= (x \ggg 19) \oplus (x \ggg 61) \oplus (x \gg 6)
\end{aligned}$$

for SHA-512.

Let $W_{i,j}$ denote bit j of W_i, where $W_{i,0}$ is the least significant bit and $W_{i,w-1}$ is the most significant bit. We also use the same notation to denote bit positions for other variables such as A_i and E_i.

2.2 Quantum Computation

We use the quantum circuit model as the model of quantum computation. Let H denote the Hadamard operator defined by $H|b\rangle = \sum_{c\in\{0,1\}}(-1)^{b\cdot c}|c\rangle$ for $b \in \{0,1\}$. The quantum oracle of a function $f : \{0,1\}^m \to \{0,1\}^n$ is the unitary operator O_f defined by $O_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$ for $x \in \{0,1\}^m$ and $y \in \{0,1\}^n$.

Grover's Algorithm. Grover's algorithm [15] is the quantum algorithm to solve the following database search problem.

Problem 1. Let $F : \{0,1\}^n \to \{0,1\}$ be a function such that $|F^{-1}(1)| > 0$. Given a (quantum) oracle access to F, find x such that $F(x) = 1$.

Let $t := |F^{-1}(1)|$. We always consider the case that $t/2^n \ll 1$. Though $O(2^n/t)$ queries are required for classical algorithms to solve the problem, Grover's algorithm solves the problem only with $O(\sqrt{2^n/t})$ quantum queries.

More precisely, suppose that there exists a quantum circuit that computes F in time T_F by using S_F qubits (i.e., the depth and width of the circuit are T_F and S_F, respectively). Then, Grover's algorithm finds a solution in time $T_F \cdot (\pi/4) \cdot \sqrt{2^n/t}$, by using $S_F + 1$ qubits.

Details of Grover's Algorithm. For a positive integer i, let $\mathsf{Grov}(F, i)$ be the quantum algorithm that runs the following procedure:

1. Prepare the initial state $|\psi_{\mathsf{init}}\rangle := H^{\otimes(n+1)} |0^n\rangle |1\rangle$.
2. Let θ be the value that satisfies $\sin^2 \theta = t/2^n$ and $0 \leq \theta \leq \pi/2$. Apply the unitary operator $Q_F := -(H^{\otimes n} \otimes I)(O_0 \otimes I)(H^{\otimes n} \otimes I)O_F$ iteratively i times on $|\psi_{\mathsf{init}}\rangle$. Here, O_F is the quantum oracle of F, and O_0 is the operator such that $O_0 |x\rangle = (-1)^{\delta_{x,0^n}} |x\rangle$ ($\delta_{x,y}$ is Kronecker's delta such that $\delta_{x,y} = 1$ if $x = y$ and $\delta_{x,y} = 0$ if $x \neq y$).
3. Measure the resulting state $Q_F^i |\psi_{\mathsf{init}}\rangle$, and output the most significant n bits.

Boyer et al. showed that, when we set the number of iterations i to be $\lfloor \pi/4\theta \rfloor$, the algorithm $\mathsf{Grov}(F, \lfloor \pi/4\theta \rfloor)$ outputs x such that $F(x) = 1$ with a probability at least $1 - t/N$ [6]. Since $\pi/4\theta \leq \pi/(4 \sin\theta) = (\pi/4)\sqrt{2^n/t}$ holds, the running time of $\mathsf{Grov}(F, \lfloor \pi/4\theta \rfloor)$ is at most $T_F \cdot (\pi/4)\sqrt{2^n/t}$.

Remark 2. In the above arguments, we implicitly assume that t is known in advance. If t is not known in advance, we have to perform a more sophisticated procedure, which increases the total number of queries to F by a constant factor [6].

Parallelization. When $P \geq 2$ quantum computers are available, by running P copies of $\mathsf{Grov}(F, \lfloor \pi/4\theta\sqrt{P} \rfloor)$ in parallel, we can find a solution in time $T_F \cdot \frac{\pi}{4}\sqrt{2^n/(t \cdot P)}$ with a probability at least $1 - 1/e$ (we always consider the case that $(t \cdot P)/2^n \ll 1$). For completeness, we provide detailed explanations on the success probability in Section C.

Cost Evaluation. As mentioned in Section 1, we evaluate the complexity of the attacks in the setting of time-space tradeoff. We do not take costs of quantum error corrections into account, and we consider that the running time of a quantum circuit is proportional to the depth of the circuit.

In each attack, we assume that there exists an implementation of the attack target primitive (i.e., SHA-256 or SHA-512) on a quantum circuit $\mathcal{C}$, and we regard that the unit of depth (resp., width) of quantum circuits is the depth (resp., width) of $\mathcal{C}$, so that our cost estimation will be independent from implementation methods of primitives.

In addition, we do not take communication costs into account. That is, we assume that arbitrary two-qubit quantum gate can be applied to arbitrary pair

of qubits. The communication costs will not be significant in our attacks because we use quantum circuits just for running the Grover search (or, its simple parallelization) that requires only several times as much qubits as implementations of SHA-2 use.

3 Previous Works

This section provides an overview on the collision attack on 31-step SHA-256 in [32], the semi-free-start collision attack on 38-step SHA-256 in [32], and the semi-free-start collision attack on 39-step SHA-512 in [10,11].

3.1 Collision Attack on 31-Step SHA-256

The collision attack on 31-step SHA-256 in [32] finds a 2-block collision with time complexity $2^{65.5}$. Intuitively, a 2-block collision $(\tilde{M}||M, \tilde{M}||M')$ (here, $\tilde{M}, M, M'$ are in $\{0,1\}^{512}$, and $M \neq M'$) is constructed by searching for a random message $\tilde{M}$ for the first block and a semi-free-start collision (M, M') for the second block such that the output of the first block is the IV of the second block.

Semi-free-start collisions in the second block are constructed based on a local collision that starts at step 5 and ends at step 18, which is found by using heuristic automated search tools. The tool finds both of differential characteristics and conditions for message pairs (M, M') at the same time. See table 2 for the differential characteristic and conditions for (M, M') shown in [32]. The meanings of the notations in Table 2 are as follows:

1. "–" indicates that the bit associated with M at the position must be equal to the corresponding bit associated with M'.
2. "0" indicates that the bit at the position must be 0 for both of M and M'.
3. "1" indicates that the bit at the position must be 1 for both of M and M'.
4. "u" indicates that the bit at the position must be 1 for M and 0 for M'.
5. "n" indicates that the bit at the position must be 0 for M and 1 for M'.

See also Remark 3. For each i, by A_i, E_i, W_i we denote the words of internal states and expanded messages as described in Section 2.1.

The authors of [32] also show an example of a semi-free-start collision of 31-step SHA-256 that satisfies the differential characteristic. See Table 6 in the appendix for details.

Attack procedure. Next, we describe the attack procedure. The important features of the differential characteristic in Table 2 are summarized as follows:

1. Only seven message words $(W_5, \ldots, W_9, W_{16}, W_{18})$ have differences. Since $W_0, \ldots, W_4, W_{10}, \ldots, W_{15}$ do not have differences, $W_{17}, W_{19}, W_{26}, \ldots, W_{30}$ do not have differences, either. The differences at $W_{20}, \ldots, W_{25}$ need to be canceled out (see Table 3).

Table 2. The 31-step differential characteristic for SHA-256 shown in [32].

i	ΔA_i	ΔE_i	ΔW_i
−4	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
−3	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
−2	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
−1	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
0	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
1	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
2	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
3	---- ---- ---- ---- ---- ---- ---- --0-	-0-- ---- ---- ---- --0- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
4	---- ---- ---- ---- ---- ---- ---- --00	-1-- ---- ---1 ---- -01- --1- 0--0 --10	---- ---- ---- ---- ---- ---- ---- ----
5	-nnn -n-n -11- ---n --nu -1-- ---- -0n-	0nnn n1uu -0-1 101n -1nu --0- 11-1 -0n1	u--- uunu ---- ---n ---n ---- ---- --n-
6	unnn n--- ---- ---- ---- ---- ---- --0-	n-n1 0111 n--u 11u0 0n10 u1n- nn1n -1uu	nn1- n--- nu-n n--1 u--0 -un0 --n0 -nn-
7	---- ---- ---- ---- ---n ---- ---- n-0u	101u 0nn1 0-11 011u -n11 1n11 0un1 -nnn	00nn 0n10 1-n1 nnn1 u0nn -n01 1u-1 n0--
8	---- ---- ---- ---- ---- ---- ---- ----	1-uu 1111 0--0 u101 10n- 1010 1010 -0n0	0001 u000 1-00 0nuu un1n 01nn -01n uuuu
9	---- ---- ---- ---- ---- ---- ---- ----	1011 00uu 1111 11nu 1110 01-- 0111 10nn	---- -1-- ---- ---u n--- 0--- --11 un--
10	---- ---- ---- ---- u--- ---- ---- -u--	1-00 u110 1001 101u n00- -000 1--u 1n00	---0 ---- ---- ---- ---- ---- ---- --1-
11	---- ---- ---- ---- ---- ---- ---- ----	0101 00u0 nu1u uuuu u100 1000 000n 1u10	---- ---- ---- ---- ---- ---- ---- ----
12	---- ---- ---- ---- ---- ---- ---- ----	111n uuuu uuuu uuuu u001 1111 0110 0n00	---- ---- ---- ---- ---- ---- ---- ----
13	---- ---- ---- ---- ---- ---- ---- ----	---1 01-1 1-1- ---- 1--- ---- ---0 -0--	---- ---- ---- ---- ---- ---- ---- ----
14	---- ---- ---- ---- ---- ---- ---- ----	---1 00-- -001 1111 u--- ---- 1--- -u--	---- ---- ---- ---- ---- ---- ---- ----
15	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- 0--- ---- ---- -0--	---- ---- ---- ---- ---- ---- ---- ----
16	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- 1--- ---- ---- -1--	---- ---- ---- -unn nunn nnnn nnnn nn--
17	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
18	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- n--- ---- ---- -n--
19	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
20	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
21	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
22	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
23	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
24	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
25	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
26	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
27	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
28	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
29	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
30	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----

2. No condition is imposed on the first five message words $W_0, \ldots, W_4$, thus those can be chosen freely.

By using these properties, the authors of [32] first show an attack with complexity $2^{99.5}$, and then show how to reduce the complexity to $2^{65.5}$.

The first attack with complexity $2^{99.5}$. Let f denote the (31-step) compression function. The procedure of the collision attack with complexity $2^{99.5}$ is as follows.

I. Use the automatic search tool to determine the message words $W_5, \ldots, W_{12}$ and the internal states from the beginning of step 5 to the end of step 12 (in the second block). Though $W_0, \ldots, W_4$ have not been chosen yet at this

Table 3. The position of the message words where non-zero differences appear. "◯" indicates that the word has non-zero difference. "×" indicates that the word is computed from previous words with non-zero differences but the difference is canceled out. (W_i is computed from W_{i-2}, W_{i-7}, W_{i-15}, and W_{i-16} for $i \geq 16$.)

W_i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Difference						◯	◯	◯	◯	◯						

W_i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Difference	◯		◯		×	×	×	×	×	×					

step, the values of the variables $E_1, \ldots, E_4$ and $A_{-3}, \ldots, A_4$ are completely determined by the internal state at the beginning of step 5 (see also Fig. 4 in the appendix materials). Note that $A_{-1}||A_{-2}||A_{-3}$ correspond to the 96 most significant bits of the initial value of the second block.

II. Find a message $\tilde{M}$ for the first block such that the 96 most significant bits of $f(\mathrm{IV}, \tilde{M})$ is equal to $A_{-1}||A_{-2}||A_{-3}$. Compute the (uniquely determined) values $W_0, \ldots, W_4$ that is compatible with the chaining value $f(\mathrm{IV}, \tilde{M})$ and the state at the beginning of step 5.

III. Now, $W_0, \ldots, W_{12}$ have been chosen. Use degrees of freedom in W_{13}, W_{14}, W_{15} to fulfill the conditions on E_{13}, E_{14}, E_{15}, W_{16}, and W_{18} (in addition to the cancellation of differences at $W_{20}, \ldots, W_{25}$). If it fails, go back to Step II.

Step II requires time 2^{96}. According to the authors of [32], Step I of the attack takes only seconds, and Step III succeeds with a probability about $1/12$ due to the lack of degrees of freedom in W_{13}, W_{14}, W_{15}, which was verified experimentally. The total time complexity is estimated as $12 \cdot 2^{96} \approx 2^{99.5}$.

The second attack with complexity $2^{65.5}$. The attack complexity is reduced from $2^{99.5}$ to $2^{65.5}$ by computing many solutions in Step I. The idea is as follows. Suppose that ℓ solutions can be found for Step I (they are stored in a list). Then, the complexity of Step II can be reduced from 2^{96} to $2^{96}/\ell$. If a single solution in Step I can be found in time T_I, then the overall complexity of the attack becomes $T_I \cdot \ell + 12 \cdot 2^{96}/\ell$.

The authors of [32] claim that $T_I \approx 2^{25.5}$, and their experiments indicate that they can expect $\ell \approx 2^{34}$. Based on these observations, they deduced that a collision can be found with complexity $2^{25.5} \cdot 2^{34} + 12 \cdot 2^{96}/2^{34} \approx 2^{65.5}$.

3.2 Semi-Free-Start Collision Attack on 38-Step SHA-256

As well as the semi-free-start collisions in the 31-step collision attack, the semi-free-start collision of 38-step SHA-256 in [32] is constructed based on a local collision that starts at step 7 and ends at step 24, which are also found by using the heuristic automated search tool.

See Table 4 for the differential characteristic and the conditions for confirming message pairs shown in [32]. (See also Remark 3.) The semi-free-start collision of 38-step SHA-256 given in [32] is shown in Table 7 in the appendix.

3.3 Semi-Free-Start Collision Attack on 39-Step SHA-512

The semi-free-start collision of 39-step SHA-512 in [10,11] is also constructed based on a local collision that starts at step 8 and ends at step 26, which is also found by using the heuristic automated search tool.

See Table 5 in the appendix for the differential characteristic and the conditions for confirming message pairs shown in [11]. (See also Remark 3.) The semi-free-start collision of 39-step SHA-512 given in [10,11] is shown in Table 8 in the appendix.

Table 4. The 38-step differential characteristic for SHA-256 shown in [32].

i	ΔA_i	ΔE_i	ΔW_i
−4	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
−3	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
−2	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
−1	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	
0	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
1	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
2	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
3	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
4	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
5	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- 1--- -1-- ----	---- ---- ---- ---- ---- ---- ---- ----
6	---- ---- ---- ---- ---- ---- ---- ----	1--0 0--1 ----- ---1 -101 11-- -0-1 --0-	---- ---- ---- ---- ---- ---- ---- ----
7	-n-- --u- n-u- --u- ----- --n- nn-- ----	nu-1 1-0u uun1 01-u uuuu uu1u -u-1 --01	--nn nnn- -nn- -un- ---n uuu- nn-- ----
8	---- nn-n --n- -un- uu-u -u-u -n-- ----	n01n n-1n 1000 0-1u 1uuu nu00 n0-n n0n1	0000 0110 1110 1--1 100n uuuu uuuu uu00
9	u--- u-n- -nuu u--- n-uu u--u ---- ---n	000n 00u1 0001 n101 nu0u 0001 11-u 11un	---- ---- ---- ---- ---- ---- ---- ----
10	---- ---- ---- ---- ---- ---- ---- ----	00un n001 10n1 001u 11u- 101u 111u 0uun	---- ---- unnn nnnn u--- ---- ---- ----
11	---- ---- ---- ---- ---- ---- ---- ----	1u1- 11u1 1-n0 1n10 0n10 u1-1 1u10 0011	---- --1- ---- ---- ---- ---- ---- ----
12	---- ---- ---- ---- ---- ---- ---- ----	0uu1 u1u0 u1uu 1n01 nn11 1u01 1n-0 1010	---- ---- --1- ---- 01-- ---- ---- ----
13	---- ---- ---- ---- ---- ---- ---- ----	n001 0uu0 1-00 n1-0 1n0n u10u 10-1 -nuu	---- ---- ---- ---- 0--- ---- ---- ---1
14	---- ---- ---- ---- ---- ---- ---- ----	101- 110- 1-00 10-1 0--1 11-0 10-- -100	---- ---- ---- ---- ---- ---- ---- ----
15	---- u--- ---- ---n ---- ---- ---- ----	0-1- u01- ---0 0--n -00- ---1 0--- -111	---- ---- ---- ---- ---- ---- ---- -u--
16	---- ---- ---- ---- ---- ---- ---- -u--	---- n-n1 ---0 1-un 11-- ---- nuu- nu01	---- ---- ---- ---- ---- ---- ---- ----
17	---- ---- ---- ---- ---- ---- ---- ----	-0-- n-1- --0n nnnn -nuu 1--- 011- 1-un	---- ---- ---- ---- ---- ---- ---- ----
18	---- ---- ---- ---- ---- ---- ---- ----	---- 0-1- --00 000- -000 --10 1110 1100	---- ---- ---- ---- ---- ---- ---- ----
19	---- ---- ---- ---- ---- ---- ---- ----	0--- u-00 nuuu uuuu 0001 --00 1101 1011	---- ---- ---- ---- ---- ---- ---- ----
20	---- ---- ---- ---- ---- ---- ---- ----	1--1 --11 1001 11-- -1-- 0unn nnnn nn0-	---- ---- ---- ---- ---- ---- ---- ----
21	---- ---- ---- ---- ---- ---- ---- ----	---- 1--- 1111 1111 ---- -000 0000 00--	---- ---- ---- ---- ---- ---- ---- ----
22	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- -111 1111 11--	---- ---- ---- ---- ---- ---- ---- ----
23	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- n--- ---- --un ---- ---- ---- ----
24	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- -n--
25	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
26	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
27	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
28	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
29	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
30	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
31	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
32	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
33	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
34	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
35	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
36	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----
37	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ----

Remark 3. To be precise some bits in the differential characteristics in Tables 2, 4, and 5 have additional conditions. They are shown in the original papers [32,10,11] but we omit to show them because they are not significantly relevant to the basic idea of our attacks.

4 Observations and Ideas for Quantum Collision Attacks

For SHA-256, the previous 38-step semi-free-start collision is not converted into a collision while the 31-step semi-free-start collision is converted. For SHA-512, the previous 39-step semi-free-start collision is not converted.

In Section 4.1, we explain details on the reason that the semi-free-start collisions of 38-step SHA-256 and 39-step SHA-512 are not converted into collisions in the classical setting, based on the explanation in [10,11]. In Section 4.2, we explain our basic ideas on how to apply the conversion in the quantum setting.

4.1 Obstacles for Conversions in the Classical Setting

Summary of 31-Step SHA-256. Recall that the 31-step collision is obtained by matching the IV produced from the first block and semi-free-start collisions in the second block. Also recall that the attack consists of three steps.

In Step II of the attack, the degrees of freedom in the message words $W_0, \ldots, W_4$ are used to make the output of the first block and the local collision in the second block compatible. Let α denote the number of free bits in the message words that can be used to make those two values compatible. Since $W_0, \ldots, W_4$ can be chosen freely, $\alpha = 5 \cdot 32 = 160$ holds. For a randomly chosen $\tilde{M}$ and a single solution in Step I, the probability that they can be compatible is $2^\alpha/2^n$.

If we have ℓ solutions in Step I and if Step III succeeds with a probability p, a randomly chosen $\tilde{M}$ leads to a collision with probability $\ell \cdot (2^\alpha/2^n) \cdot p$. Thus the time complexity T is estimated as $T = \frac{1}{\ell \cdot (2^\alpha/2^n) \cdot p} = 2^n/(\ell \cdot 2^\alpha \cdot p)$ (by ignoring the complexity of Step I).

It is claimed in [32] that one can expect $\ell = 2^{34}$ and $p \approx 1/12 \approx 2^{-3.5}$, and the complexity $2^{65.5}$ is obtained as $T = 2^{256}/(2^{34} \cdot 2^{160} \cdot 2^{-3.5}) = 2^{65.5}$.

Remark 4. Let 2^X be the number of IVs of the second block that will be compatible with the local collisions in the second block. Then, the time complexity to find the first message block will be $T = 2^n/2^X$. The attack is valid as long as $X > n/2 = 128$.

Lack of Degrees of Freedom in 38-Step SHA-256. We observe that in the differential characteristic for the 38-step semi-free-start collision of SHA-256 (Table 4), almost all the bits of state variable E_i have conditions for $i = 7, \ldots, 20$, which implies that both of the values and the differences for $W_7, \ldots, W_{20}$ will be fixed. When 16 successive message words are fixed, all the message words are fixed (due to the message expansion). Thus, among the message words $W_0, \ldots, W_7$ that can be used to make the first block and the local collision in the second block compatible, only the two words W_5 and W_6 will have degrees of freedom, and the number of free bits is $\alpha = 2 \cdot 32 = 64$ in total.

Thus the time complexity will be $2^n/(\ell \cdot 2^\alpha \cdot p) = 2^{192}/(\ell \cdot p)$ when ℓ solutions are available. Considering that ℓ is about 2^{34} for 31-step collisions, the complexity will be larger than 2^{128} of the birthday paradox.

Remark 5. From another point of view, the 38-step semi-free-start collision cannot be converted into a collision because $X < 128$.

Lack of Degrees of Freedom in 39-Step SHA-512. The 39-step semi-free-start collision of SHA-512 in [32,11] cannot be converted into a collision for the same reason.

In the differential characteristic (Table 5), almost all bits of the internal state variable E_i have some conditions for $i = 8, \ldots, 22$, which implies that both of the internal states and the message words in steps 8 - 22 will be fixed. Due to the constraint derived from the message expansion, only the single word W_7 will have degrees of freedom among the first 8 message words that can be used to make the first block and the local collision in the second block compatible (i.e., $\alpha = 64$). In addition, ℓ will not be large since the differential characteristic has dense conditions for $i = 8, \ldots, 22$. Thus the time complexity $2^n/(\ell \cdot 2^\alpha \cdot p)$ will be larger than 2^{256} of the birthday paradox.

4.2 Observations and Ideas on Conversion in the Quantum Setting

As mentioned in Remark 4, $X > n/2$ must be satisfied to be a valid attack. On the other hand, in the quantum setting of time-space tradeoff, it may be possible to mount valid 2-block collision attacks even if $X < n/2$. For example, assume that we can decrease the time complexity of 2-block collision attacks from $2^n/2^X$ to $\sqrt{2^n/2^X}$ by applying the Grover search and the Grover search requires negligible memory. It becomes a valid quantum collision attack in the setting of time-space tradeoff if $\sqrt{2^n/2^X} < 2^{n/2}$, which is equivalent to $X > 0$. Actually this idea is too naive and we cannot achieve a valid quantum attack in such a simple way. Nevertheless, this idea shows the possibility of valid 2-block quantum collision attacks with the Grover search.

With this in mind, we mount quantum collision attacks on 38-step SHA-256 and 39-step SHA-512 by converting the semi-free-start collisions into 2-block collisions with the Grover search. To achieve this goal, we have to take the following two points into account.

1. In the classical attack on 31-step SHA-256, by storing ℓ solutions in Step I, the complexity of Step II is decreased by the factor of ℓ. This strategy works well since memory is relatively cheap in the classical setting. On the other hand, memory is expensive in the quantum setting of time-space tradeoff, and memory-less algorithms are favorable.
2. In the classical attack on 31-step SHA-256, we can choose $W_0, \ldots, W_4$ freely because those values do not affect the steps with dense conditions in the differential characteristic (i.e., steps 5 - 12). On the other hand, in the attack on 38-step SHA-256 (resp., 39-step SHA-512), we have to choose the message words $W_0, \ldots, W_6$ (resp., $W_0, \ldots, W_7$) carefully because they affect on some of the message words in the steps with dense conditions, i.e., $W_7, \ldots, W_{20}$ (resp., $W_8, \ldots, W_{22}$), through the message expansion.

We will set $\ell = 1$ to minimize the required memory size. On the choice of the message words $W_0, \ldots, W_6$ for 38-step SHA-256, we observe the following.

> We can modify W_6 to another value $\hat{W}_6$ without changing $W_7, \ldots, W_{21}$ by modifying W_j to $\hat{W}_j := W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ for $j = 5, 4, \ldots, 0$ step by step.

Indeed, if the value of W_6 is changed to another value $\hat{W}_6$, then W_{21} and W_{22} will be changed because $W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16}$ holds for $i \geq 16$. However, the change of the value of W_{21} can be canceled out by modifying W_5 to $\hat{W}_5 := W_5 - (\sigma_0(\hat{W}_6) - \sigma_0(W_6))$. By modifying W_j to $\hat{W}_j := W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ similarly for $j = 4, \ldots, 0$, we can also keep $W_{20}, \ldots, W_{16}$ unchanged. Since $W_7, \ldots, W_{15}$ are not affected by the modification of $W_0, \ldots, W_6$, the words $W_7, \ldots, W_{15}$ are also kept unchanged.

We obtain a similar observation on the choice of the message words $W_0, \ldots, W_7$ for 39-step SHA-512. That is, we can modify W_7 to another value $\hat{W}_7$ without changing $W_8, \ldots, W_{22}$, by modifying W_j to $\hat{W}_j := W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ for $j = 6, \ldots, 0$ step by step.

We mount quantum 2-block collision attacks based on these observations.

Attack Idea. Here we explain basic ideas of our quantum attacks that are common between 38-step SHA-256 and 39-step SHA-512. We will explain details that are specific to each function in the next section.

Let i denote the number of the step where the local collision starts in the differential characteristic ($i = 7$ for 38-step SHA-256 and $i = 8$ for 39-step SHA-512). The attack procedure is as follows (see also Fig. 5 in the appendix).

I. Find a pair of messages (M, M') and an initial value for the second block that yield a semi-free-start collision. Let S_{start} be the internal state at the beginning of step i. For each j, let W_j and W'_j denote message word j expanded from M and M', respectively. Note that $W_0 = W'_0, \ldots, W_{i-1} = W'_{i-1}$ hold.

II. With the Grover search, find a message $\tilde{M}$ (for the first block) that satisfies the followings.
   (a) S_{start} and the input chaining value for the second block $\text{IV}_{\text{second}}$ derived from $\tilde{M}$ can be compatible by modifying the message words $W_0, \ldots, W_{i-1}$, $W'_0, \ldots, W'_{i-1}$, while keeping the message words $W_i, \ldots, W_{i+14}, W'_i, \ldots, W'_{i+14}$ unchanged. Let $\hat{M}$ and $\hat{M}'$ be the messages for the second block after the modification, i.e., $\hat{M} := \hat{W}_0 || \cdots || \hat{W}_{i-1} || W_i || \cdots || W_{15}$ and $\hat{M}' := \hat{W}_0 || \cdots || \hat{W}_{i-1} || W'_i || \cdots || W'_{15}$.
   (b) S_{start} and the modified message pair $(\hat{M}, \hat{M}')$ yield a collision at the end of the second block.

III. By using $\tilde{M}$ found in Step II, perform the computations in Steps II-(a) and II-(b) again to obtain the pair $(\hat{M}, \hat{M}')$ that yield a collision at the end of the second block. (This step may seem redundant, but we separate this step from Step II so that we can apply the Grover search on $\tilde{M}$ in Step II.) Output $(\tilde{M}||\hat{M}, \tilde{M}||\hat{M}')$.

Step I of the above procedure corresponds to Step I of the classical collision attack on 31-step SHA-256. We store only a single solution in Step I of our attack so that the attack will be memory-less. Since only a single solution is required in this step, we just use the values shown in the previous works (i.e., the values M, M', h_0 in Table 7 and Table 8 in the appendix).

Step II-(a) corresponds to Step II of the classical collision attack on 31-step SHA-256. Step II-(b) corresponds to Step III of the classical collision attack on 31-step SHA-256. We allow the remaining words $W_{i+15}, W_{i+16}, \ldots$ and $W'_{i+15}, W'_{i+16}, \ldots$ to be changed since the steps with dense conditions are up to $i + 14$ and thus to probabilistically satisfy all the conditions from step $i + 15$ by randomly changed $W_{i+15}, W_{i+16}, \ldots$ and $W'_{i+15}, W'_{i+16}, \ldots$ is not difficult.

Attack Complexity and Validity. Let F be the Boolean function to which Grover's algorithm is applied in Step II of our attack[8]. That is, F is defined by $F(\tilde{M}) := 1$ if and only if $\tilde{M}$ satisfies the two conditions II-(a) and II-(b). Let p be the probability that $F(\tilde{M}) = 1$ when we pick a message $\tilde{M}$ for the first block uniformly at

[8] More precisely, we run $\mathsf{Grov}(F, \lfloor \pi/4\theta \rfloor)$ in Step II, where $\theta = \arcsin(\sqrt{p})$.

random. In addition, suppose that F can be implemented on a quantum circuit of which width is S_F and depth is T_F.

The time complexity of Step I is negligible since we just use the values from previous works. The time complexity of Step III is also negligible compared to that of Step II. Thus the time complexity of our attacks is dominated by the time complexity of the Grover search on F, which is at most $T_F \cdot \frac{\pi}{4}\sqrt{1/p}$.

If a quantum computer of size $S(> S_F)$ is available, the Grover search can be parallelized[9] and sped up by the factor of $\sqrt{S/S_F}$, and the attack time complexity becomes

$$\left(T_F \cdot (\pi/4)\sqrt{1/p}\right) / \sqrt{S/S_F} = T_F \cdot (\pi/4) \cdot \sqrt{S_F/pS}. \tag{1}$$

Let n be the output length of the hash function. Since the time complexity of the generic attack is $2^{n/2}/S$ when a quantum computer of size S is available, our attack is valid as long as

$$T_F \cdot (\pi/4) \cdot \sqrt{S_F/pS} < 2^{n/2}/S \tag{2}$$

holds.

Remark 6. When we run the same procedure in the classical setting, the Grover search is replaced with the usual exhaustive search and the attack time complexity will be $(T_F \cdot S_F)/p$ (here we do not consider parallelizations for simplicity). Since the generic complexity is $2^{n/2}$, the attack becomes valid if and only if $(T_F \cdot S_F)/p < 2^{n/2}$, which is equal to

$$p > (T_F \cdot S_F)/2^{n/2}. \tag{3}$$

In particular, the classical attack is invalid if $p < 2^{-n/2}$. On the other hand, the condition (2) is equivalent to $p > S_F \cdot (\pi^2/16) \cdot T_F^2/2^n$ (when $S = 1$), and the quantum attack may be valid even if $p < 2^{-n/2}$.

5 Quantum Collision Attack on 38-Step SHA-256

This section shows a quantum collision attack on 38-step SHA-256 based on the attack idea in Section 4.2.

Let (M, M') and h_0 be the semi-free-start collision and the initial value shown in the previous work (i.e., (M, M') and h_0 in Table 7 in the appendix). Let W_j and W'_j denote message word j associated with M and M', respectively. Recall that the local collision starts at step 7 in the differential characteristic in Table 4. Let S_{start} be the state at the beginning of step 7 that is computed from (M, M') and h_0.

Section 5.1 provides some observations on Step II of the quantum attack. Section 5.2 provides an implementation of F and analyzes the depth and width of the circuit of F. In Section 5.3 we analyze the total complexity when the quantum attack is mounted with the implementation of F in Section 5.2.

[9] See Section 2 for details on parallelization. We use the quantum computer of size S as S/S_F independent small quantum computers.

5.1 Observation on Step II

We provide two observations.

First Observation. The internal state variables $A_{-1}, \ldots, A_6$ and $E_3, \ldots, E_6$ are determined from $S_{\text{start}} = A_6 || \cdots || A_3 || E_6 || \cdots || E_3$. (These variables are common between M and M'. See also Fig. 4 in the appendix.) There exists a tuple $(\hat{W}_0, \ldots, \hat{W}_6)$ that is compatible with $\text{IV}_{\text{second}}$ and S_{start} if and only if A_{-1} matches the most significant 32 bits of $\text{IV}_{\text{second}}$. If A_{-1} matches, A_{-2}, A_{-3}, A_{-4}, $E_{-1}, \ldots, E_{-4}$ are determined by the equation $\text{IV}_{\text{second}} = A_{-1} || \cdots || A_{-4} || E_{-1} || \cdots || E_{-4}$, and the message words $\hat{W}_0, \ldots, \hat{W}_6$ are uniquely determined from $A_{-4}, \ldots, A_6$ and $E_{-4}, \ldots, E_{-1}, E_3, \ldots, E_6$.

Second Observation. By exhaustively checking all the possible values for $\hat{W}_6 \in \{0,1\}^{32}$, we verified that there exist 1179647 ($> 2^{20}$) tuples $(\hat{W}_0, \ldots, \hat{W}_6)$ that satisfy the following conditions.[10]

(i) $\hat{W}_j = W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ holds for $j = 0, \ldots, 5$.
(ii) S_{start} and the messages $(\hat{M}, \hat{M}')$ for the second block, where $\hat{M} := \hat{W}_0 || \cdots || \hat{W}_6 || W_7 || \cdots || W_{15}$ and $\hat{M}' := \hat{W}_0 || \cdots || \hat{W}_6 || W'_7 || \cdots || W'_{15}$, yield a collision at the end of the second block.

Remark 7. From another point of view, the second observation shows that we can make semi-free-start collisions for at least 2^{20} initial values.

5.2 Implementation and Analysis of F

Below we provide an implementation of F and its analysis. In particular, we show $T_F \leq 6.8$ and $S_F \leq 3.9$, where S_F denotes the width of the quantum circuit of F and T_F denotes the running time (depth) of the circuit.

Implementation of F: Basic Idea. Before describing a formal implementation of F with notations of quantum computation, we give a basic idea behind the implementation.

First, we compute the following values (from Table 7 in the appendix) and store them into memory.

(a) The internal state S_{start} at the beginning of step 7.
(b) The message words $W_0 = W'_0, \ldots, W_6 = W'_6, W_7, \ldots, W_{21}, W'_7, \ldots W'_{21}$.
(c) The internal state variable A_{-1} that is uniquely determined from S_{start}.

Note that these values are computed and stored before the start and kept unchanged throughout the attack.

Given an input $\tilde{M}$, the output value $F(\tilde{M})$ is computed as follows.

[10] We actually implemented to count the number of semi-free-start collisions for all 2^{32} choices of W_6 and accordingly modified $W_5 \ldots, W_0$.

1. Compute the output of the first block from $\tilde{M}$, and let $\mathrm{IV}_{\mathrm{second}}$ denote the output.
2. Check if the condition that the most significant 32 bits of $\mathrm{IV}_{\mathrm{second}}$ is equal to A_{-1} is satisfied. If it is satisfied, proceed to the next step. Otherwise output 0 and abort.
3. Compute the unique $(\hat{W}_0, \ldots, \hat{W}_6)$ that is compatible with $\mathrm{IV}_{\mathrm{second}}$ and S_{start}.
4. Check if the following conditions are satisfied.
   (i) $\hat{W}_j = W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ holds for $j = 0, \ldots, 5$.
   (ii) S_{start} and the messages $(\hat{M}, \hat{M}')$ yield a collision at the end of the second block, where $\hat{M} := \hat{W}_0 || \cdots || \hat{W}_6 || W_7 || \cdots || W_{15}$ and $\hat{M}' := \hat{W}_0 || \cdots || \hat{W}_6 || W'_7 || \cdots || W'_{15}$.

   If both of (i) and (ii) are satisfied, output 1. Otherwise output 0.

Remark 8. When we implement a quantum circuit, each computational step has to be reversible, and the running time of the circuit has to be independent from inputs. We ignored such properties in the above explanations for simplicity but they are taken into account in the formal description below.

Implementation of F: Formal Description. Let L be the list to store the values explained in (a)-(c) above, and f be the 38-step compression function. Given an input $\tilde{M}$, the output value $F(\tilde{M})$ is computed as follows.

0. At the beginning, the quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle$. ($|y\rangle$ is the single qubit register where the value $F(\tilde{M})$ will be added.)
1. Compute the output of the first block from $\tilde{M}$. Let $\mathrm{IV}_{\mathrm{second}}$ denote the output. Check if A_{-1} is equal to the most significant 32 bits of $\mathrm{IV}_{\mathrm{second}}$. If they are equal, set $b := 1$. If they are not equal, set $b := 0$. The current quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |b\rangle$.
2. Let $\mathrm{IV}'_{\mathrm{second}}$ denote the concatenation of A_{-1} and the least significant 224 bits of $\mathrm{IV}_{\mathrm{second}}$ ($\mathrm{IV}'_{\mathrm{second}} = \mathrm{IV}_{\mathrm{second}}$ holds if $b = 1$). Compute the unique $(\hat{W}_0, \ldots, \hat{W}_6)$ that is compatible with the initial chaining value $\mathrm{IV}'_{\mathrm{second}}$ and S_{start}. The current quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |b\rangle |\hat{W}_0, \ldots, \hat{W}_6\rangle$.
3. Let $\hat{M}$ denote $\hat{W}_0 || \cdots || \hat{W}_6 || W_7 || \cdots || W_{15}$ and $\hat{M}'$ denote $\hat{W}_0 || \cdots || \hat{W}_6 || W'_7 || \cdots || W'_{15}$. Compute the values $f(\mathrm{IV}'_{\mathrm{second}}, \hat{M})$, $f(\mathrm{IV}'_{\mathrm{second}}, \hat{M}')$. The current quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |b\rangle |\hat{W}_0, \ldots, \hat{W}_6\rangle |f(\mathrm{IV}'_{\mathrm{second}}, \hat{M})\rangle |f(\mathrm{IV}'_{\mathrm{second}}, \hat{M}')\rangle$.
4. Recall that $F(\tilde{M}) = 1$ if and only if $b = 1$ and the following (i) and (ii) hold.
   (i) $f(\mathrm{IV}'_{\mathrm{second}}, \hat{M}) = f(\mathrm{IV}'_{\mathrm{second}}, \hat{M}')$.
   (ii) $\hat{W}_j = W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ holds for $j = 0, \ldots, 5$.

   Compute $F(\tilde{M})$ by checking if $b = 1$, and (i) and (ii) hold, and add the value $F(\tilde{M})$ to the $|y\rangle$ register. The current quantum state is $|\tilde{M}\rangle |L\rangle |y \oplus F(\tilde{M})\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |b\rangle |\hat{W}_0, \ldots, \hat{W}_6\rangle |f(\mathrm{IV}'_{\mathrm{second}}, \hat{M})\rangle |f(\mathrm{IV}'_{\mathrm{second}}, \hat{M}')\rangle$.
5. Uncompute Steps 1-3 to obtain $|\tilde{M}\rangle |L\rangle |y \oplus F(\tilde{M})\rangle$.

Analysis. We regard that the unit of depth (resp., width) of quantum circuits is the depth (resp., width) required to implement 38-step SHA-256 that takes 1-block inputs. In particular, we regard that the depth required to compute a single step of SHA-512 is equal to 1/38. Since the input length of 1-block SHA-256 is 512 bits and the output length is 256 bits, at least $512+256 = 768$ qubits are required to implement the function on a quantum circuit.

Depth (T_F). Step 1 of the implementation computes the compression function once. The depth required to Step 2 is 7/38 since the message words in the first 7 steps in the second block are computed in Step 2. Step 3 computes the compression function twice. The cost of Step 4 is dominated by the computation for (ii), of which cost is at most 6 steps of SHA-256. Thus the depth required for Step 4 is at most 6/38. In summary, the depth required to implement Steps 1-4 is $1 + 7/38 + 2 + 6/38 \leq 3.4$. Since we have to perform uncomputations in Step 5, we have $T_F \leq 3.4 \times 2 = 6.8$.

Width (S_F). The length of $\tilde{M}$ is 16 words. L contains data of $8+(7+15+15)+1 = 46$ words in total. y is a single bit. Thus, $(16 + 46) \times 32 + 1 = 62 \times 32 + 1$ qubits are used in Step 0 of the implementation. Step 1 requires additional $8 \times 32 + 1$ qubits to store $\mathrm{IV}_{\mathrm{second}}$ and b. Step 2 requires additional 7×32 qubits to store $\hat{W}_0, \ldots, \hat{W}_6$. Step 3 requires additional $(8 + 8) \times 32 = 16 \times 32$ qubits to store $f(\mathrm{IV}_{\mathrm{second}}, \hat{M})$ and $f(\mathrm{IV}_{\mathrm{second}}, \hat{M}')$. Therefore, to store intermediate values shown in the above implementation, $(62+8+7+16) \times 32+2 = 2978$ qubits are used in total. Hence we have $S_F \leq 2978/768 \leq 3.9$.

Remark 9. On the estimation of the width S_F, more ancilla qubits may be required to compute the intermediate variables (such as $\mathrm{IV}_{\mathrm{second}}$) used in the implementation of F. However, we expect that they will be as much ancilla qubits as required to implement 1-block 38-step SHA-256. In particular, we expect that the ratio between the number of qubits to implement F and the number of qubits to implement 1-block 38-step SHA-256 will be about 3.9, even if we take the ancilla qubits to compute the intermediate variables into account.

Remark 10. Note that we could remove $|L\rangle$ from the computation since L is a list of classical data and computations that depend on L can be executed by classically controlling the gates. However, this has no consequence on the Time-memory tradeoff sine it is just converting qubits into classical bits.

5.3 Total Complexity

This section analyzes the total complexity when the quantum attack in Section 4.2 is mounted with the implementation of F in Section 5.2.

Let p denote the probability that $F(\tilde{M}) = 1$ holds when $\tilde{M}$ is randomly chosen. $F(\tilde{M}) = 1$ holds if and only if $\tilde{M}$ satisfies the conditions in the second and fourth steps of the implementation of F. A random $\tilde{M}$ satisfies the condition in the second step with probability 2^{-32}. From the observation on Step

II in Section 5.1, (i) and (ii) in the fourth step are satisfied with probability $\left(1179647/(2^{32})^7\right) > 2^{20}/2^{224}$. Therefore

$$p = 2^{-32} \cdot \left(1179647/(2^{32})^7\right) > 2^{-32} \cdot \left(2^{20}/2^{224}\right) = 2^{-236} \tag{4}$$

holds.

The attack time complexity can be computed as in Eq. (1). We showed $T_F \leq 6.8$ and $S_F \leq 3.9$ in Section 5.2, and $p > 2^{-236}$ in (4). Therefore, when a quantum computer of size S is available, our attack finds a collision in time $6.8 \cdot (\pi/4)\sqrt{3.9/(2^{-236} \cdot S)} = \frac{6.8\pi\sqrt{3.9}}{4} \cdot 2^{118}/\sqrt{S} \leq 2^{122}/\sqrt{S}$. In addition, the attack time complexity $2^{122}/\sqrt{S}$ is lower than the generic complexity $2^{128}/S$ when $S < 2^{12}$. Therefore our attack is valid as long as $3.9 \leq S < 2^{12}$.

Remark 11. Some may consider that our complexity analysis is invalid since the first equality in (4) holds only if the output distribution of the first block is exactly equal to the uniform distribution over $\{0,1\}^{256}$, which will not be the case for 38-step SHA-256. However, still we can reasonably expect that the analysis is valid. See Section D in the appendix for details.

6 Quantum Collision Attack on 39-Step SHA-512

This section shows a quantum collision attack on 39-step SHA-512 based on the attack idea in Section 4.2.

Let (M, M') and h_0 be the semi-free-start collision and the initial value shown in the previous work (i.e., (M, M') and h_0 in Table 8). Let W_j and W'_j denote message word j associated with M and M', respectively.

The difference between the attack on 39-step SHA-512 from the one on 38-step SHA-256 is summarized as follows.

1. The local collision starts from step 8 but not step 7 (we denote the internal state at the beginning of step 8 by S_{start}).
2. The probability p $(= |F^{-1}(1)|/2^{512})$ satisfies $p > 2^{-498.4}$.
3. The implementation of F satisfies $T_F \leq 6.8$ and $S_F \leq 4.1$.

The attack finds a collision in time $2^{252.7}/\sqrt{S}$, which is valid when $4.1 < S < 2^{6.6}$.

Section 6.1 provides some observations on Step II of the quantum attack. Section 6.2 provides an implementation of F and analyzes the depth and width of the circuit of F. In Section 6.3 we analyze the total complexity when the quantum attack is mounted with the implementation of F in Section 6.2.

6.1 Observation on Step II

We provide two observations.

First Observation. Given a chaining initial input value $\text{IV}_{\text{second}}$, *there always exists a unique tuple* $(\hat{W}_0, \ldots, \hat{W}_7)$ that is compatible with $\text{IV}_{\text{second}}$ and S_{start}. This is because the local collision starts at step 8 in the differential characteristic for 39-step SHA-512 (see also Fig. 4 in the appendix).

Second Observation. We experimentally verified that there exist 13184 ($> 2^{13.6}$) tuples $(\hat{W}_0, \ldots, \hat{W}_7)$ that satisfies the following conditions.

(i) $\hat{W}_j = W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ holds for $j = 0, \ldots, 6$.
(ii) S_{start} and the messages $(\hat{M}, \hat{M}')$, where $\hat{M} := \hat{W}_0 || \cdots || \hat{W}_7 || W_8 || \cdots || W_{15}$ and $\hat{M}' := \hat{W}_0 || \cdots || \hat{W}_7 || W'_8 || \cdots || W'_{15}$, yield a collision at the end of the second block.
(iii) $\hat{W}_{23,j} = W_{23,j}$ for $j = 5, \ldots, 29$, where $\hat{W}_{23,j}$ and $W_{23,j}$ are bit j of message word 23 derived from $\hat{M}$ and M, respectively.

The condition (iii) is added to decrease the search space for $\hat{W}_7$. We chose bit 5, bit 6, ..., bit 29 because the differential characteristic (Table 5 in the appendix) has strict conditions on these bit positions of E_{23}.

Remark 12. From another view of point, the second observation shows that we can make semi-free-start collisions for at least $2^{13.6}$ initial values.

6.2 Implementation and Analysis of F

In what follows we provide description and analysis of the implementation of F used in the attack on 39-step SHA-512 and show $T_F \leq 6.8$ and $S_F \leq 4.1$.

Implementation of F: Basic Idea. Since the basic idea of the implementation is similar to that for 38-step SHA-256 in Section 5.2, here we only provide the difference from Section 5.2.

In the attack on 39-step SHA-512, there always exists a unique tuple $(\hat{W}_0, \ldots, \hat{W}_7)$ that is compatible with $\text{IV}_{\text{second}}$ and S_{start} for arbitrary $\text{IV}_{\text{second}}$ due to the first observation in Section 6.1. Therefore we skip the step (in the implementation of F in Section 5.2) to check if A_{-1} is equal to the most significant 32 bits of $\text{IV}_{\text{second}}$.

Let $\tilde{M}$ be a message for the first block, and $\text{IV}_{\text{second}}$ be the initial vector for the second block that is computed from $\tilde{M}$. We define $F(\tilde{M}) := 1$ if and only if the conditions (i) - (iii) in the second observation in Section 6.1 are satisfied for the unique tuple $(\hat{W}_0, \ldots, \hat{W}_7)$ that is compatible with $\text{IV}_{\text{second}}$ and S_{start}.

Formal Implementation of F. First, we compute the following values (from Table 8 in the appendix) and store them into a list L.

(a) The internal state S_{start} at the beginning of step 8.
(b) The message words $W_0 = W'_0, \ldots, W_7 = W'_7, W_8, \ldots, W_{22}, W'_8, \ldots W'_{22}, W_{23}$.

Given an input $\tilde{M}$, the output value $F(\tilde{M})$ is computed as follows.

0. At the beginning, the quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle$. ($|y\rangle$ is the single qubit register where the value $F(\tilde{M})$ will be added.)
1. Compute the output of the first block from $\tilde{M}$. Let $\text{IV}_{\text{second}}$ denote the output. The current quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle \otimes |\text{IV}_{\text{second}}\rangle$.

2. Compute the unique $(\hat{W}_0, \ldots, \hat{W}_7)$ that is compatible with $\mathrm{IV}_{\mathrm{second}}$ and S_{start}. The current quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |\hat{W}_0, \ldots, \hat{W}_7\rangle$.
3. Let $\hat{M}$ denote $\hat{W}_0 || \cdots || \hat{W}_7 || W_8 || \cdots || W_{15}$ and $\hat{M}'$ denote $\hat{W}_0 || \cdots || \hat{W}_7 || W'_8 || \cdots || W'_{15}$. Compute $f(\mathrm{IV}_{\mathrm{second}}, \hat{M})$, $f(\mathrm{IV}_{\mathrm{second}}, \hat{M}')$, and $\hat{W}_{23}$, where $\hat{W}_{23}$ is word 23 derived from $\hat{M}$. The current quantum state is $|\tilde{M}\rangle |L\rangle |y\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |\hat{W}_0, \ldots, \hat{W}_7\rangle |f(\mathrm{IV}_{\mathrm{second}}, \hat{M})\rangle |f(\mathrm{IV}_{\mathrm{second}}, \hat{M}')\rangle |\hat{W}_{23}\rangle$.
4. Recall that $F(\tilde{M}) := 1$ if and only if the following (i)-(iii) hold.
   (i) $f(\mathrm{IV}_{\mathrm{second}}, \hat{M}) = f(\mathrm{IV}_{\mathrm{second}}, \hat{M}')$.
   (ii) $\hat{W}_j = W_j - (\sigma_0(\hat{W}_{j+1}) - \sigma_0(W_{j+1}))$ holds for $j = 0, \ldots, 6$.
   (iii) $\hat{W}_{23,j} = W_{23,j}$ holds for $j = 5, \ldots, 29$
   Compute $F(\tilde{M})$ by checking if (i) - (iii) hold, and add the value $F(\tilde{M})$ to the $|y\rangle$ register. The current quantum state is $|\tilde{M}\rangle |L\rangle |y \oplus F(\tilde{M})\rangle \otimes |\mathrm{IV}_{\mathrm{second}}\rangle |\hat{W}_0, \ldots, \hat{W}_7\rangle |f(\mathrm{IV}_{\mathrm{second}}, \hat{M})\rangle |f(\mathrm{IV}_{\mathrm{second}}, \hat{M}')\rangle |\hat{W}_{23}\rangle$.
5. Uncompute Steps 1-3 to obtain $|\tilde{M}\rangle |L\rangle |y \oplus F(\tilde{M})\rangle$.

Analysis. We regard that the unit of depth (resp., width) of quantum circuits is the depth (resp., width) required to implement 39-step SHA-512 that takes 1-block inputs. In particular, we regard that the depth required to compute a single step of SHA-512 is equal to 1/39. Since the input length of 1-block SHA-512 is 1024 bits and the output length is 512 bits, at least $1024 + 512 = 1536$ qubits are required to implement the function on a quantum circuit.

Depth (T_F). Step 1 of the implementation computes the compression function once. Since the message words in the first 8 steps in the second block are computed in Step 2, the depth required for Step 2 is 8/39. Step 3 computes the compression function twice. The cost of Step 4 is dominated by the computation for (ii), which is at most 7 steps of SHA-512. Thus the depth required for Step 4 is at most 7/39. In summary, the depth required to implement Steps 1-4 is $1 + 8/39 + 2 + 7/39 \leq 3.4$. Since we have to perform uncomputations in Step 5, we have $T_F \leq 3.4 \times 2 = 6.8$.

Width (S_F). The length of $\tilde{M}$ is 16 words. L contains data of $8+(8+15+15+1) = 47$ words in total. y is a single bit. Thus, $(16+47) \times 64 + 1 = 63 \times 64 + 1$ qubits are used in Step 0 of the implementation. Step 1 requires additional 8×64 qubits to store $\mathrm{IV}_{\mathrm{second}}$. Step 2 requires additional 8×64 qubits to store $\hat{W}_0, \ldots, \hat{W}_7$. Step 3 requires additional $(8 + 8 + 1) \times 64 = 17 \times 64$ qubits to store $f(\mathrm{IV}_{\mathrm{second}}, \hat{M})$, $f(\mathrm{IV}_{\mathrm{second}}, \hat{M}')$, and $\hat{W}_{23}$. Therefore, to store intermediate values shown in the above implementation, $(63 + 8 + 8 + 17) \times 64 + 1 = 6145$ qubits are used in total. Hence we have $S_F \leq 6145/1536 \leq 4.1$.

Remark 13. On the estimation of the width S_F, more ancilla qubits may be required to compute the intermediate variables (such as $\mathrm{IV}_{\mathrm{second}}$) used in the implementation of F. However, we expect that they will be as much as ancilla qubits required to implement 2-block 39-step SHA-512. In particular, we expect that the ratio between the number of qubits to implement F and the number of qubits to implement 2-block 39-step SHA-512 will be as much as 4.1, even if we take the ancilla qubits to compute the intermediate variables into account.

6.3 Total Complexity

Let p denote the probability that $F(\tilde{M}) = 1$ holds when $\tilde{M}$ is randomly chosen. $F(\tilde{M}) = 1$ holds if and only if the conditions (i)- (iii) in the fourth step of the implementation of F are satisfied. From the second observation on Step II in Section 6.1, (i)-(iii) are satisfied with probability at least $2^{13.6}/(2^{64})^8$. Therefore $p > 2^{13.6}/(2^{64})^8 = 2^{-498.4}$ holds.

The attack time complexity can be computed as in Eq. (1). We showed $T_F \leq 6.8$ and $S_F \leq 4.1$ in Section 6.2, and $p > 2^{-498.4}$ above. Therefore, when a quantum computer of size S is available, our attack finds a collision in time $6.8 \cdot (\pi/4)\sqrt{4.1/(2^{-498.4} \cdot S)} = \frac{6.8\sqrt{4.1}\pi}{4} \cdot 2^{249.2}/\sqrt{S} \leq 2^{252.7}/\sqrt{S}$. In addition, the attack time complexity $2^{252.7}/\sqrt{S}$ is lower than the generic complexity $2^{256}/S$ when $S < 2^{6.6}$. Therefore our attack is valid as long as $4.1 \leq S < 2^{6.6}$.

7 Discussion

The previous sections exploited the existing semi-free-start collision attacks to mount quantum collision attacks for SHA-256 and SHA-512. This brings the following two questions. First, is it possible to optimize differential characteristics for the classical semi-free-start collision attack with respect to the conversion to the quantum collision attack? Second, is it possible to extend the conversion framework so that a wider class of the classical attack on other computation structure can be converted into a quantum collision attack? This section answers those questions. We hope those will provide future researchers with useful knowledge to find new quantum collision attacks.

7.1 Towards Searching for New Semi-Free-Start Collision Attacks

The attacks on SHA-256 and SHA-512 in Sections 5 and 6 directly used the differential characteristics from the previous works, but it is possible to search for new differential characteristics from scratch in future works to be optimized in our conversion. More importantly, differential characteristics that cannot be exploited in the classical setting may still be exploited in the quantum setting.

Properties Required for Differential Characteristics. Our conversion is applied when the differential characteristic for the semi-free-collision attack satifies the following properties.

- The characteristic is dense, i.e. requiring many conditions, only in a relatively small number of steps. Let FIX_{start} and FIX_{end} be the input and output state values of these steps, respectively.
- For multiple choices of IV_{second}, it is possible to modify message words W_0 to W_{s-1} so that IV_{second} and FIX_{start} are connected.
- The probability to satisfy the characteristic from FIX_{end} is high enough to be faster than the generic attack.

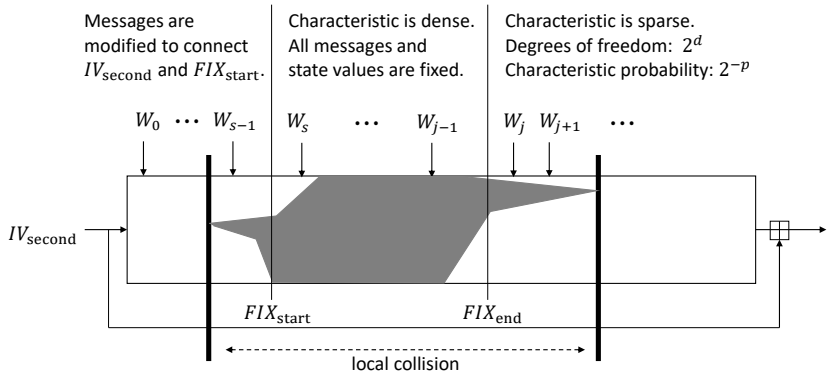

Fig. 3. Form of Semi-free-start Collision Attacks that can be Converted into Collisions.

Given those, we can view that the characteristic is composed of three parts as shown in Fig. 3. [11]

Properties for the Sparse Part. In our attacks on SHA-256 and SHA-512, (almost) all the message words are fixed after modifying W_0 to W_{s-1}, thus degrees of freedom to satisfy the characteristic from FIX_{end} to the end is provided by generating the first message block many times, which requires significant computational cost. Besides, the probability from FIX_{end} is reasonably high, thus the attack procedure could be provided without special attention. Here we give a decent analysis with respect to the condition to be faster than the generic attack, which should be taken into account for finding new characteristic.

Suppose that the first k message words are independently chosen and the dense part of the characteristic is located between step s and $j-1$, where $0 < s < j < k$. Then, after modifying W_0 to W_{s-1} to connect IV_{second} and FIX_{start}, the attacker can still have degrees of freedom in message words W_j to W_{k-1}. Let 2^d and 2^{-p} be the amount of degrees of freedom available to the attacker and the probability of the differential characteristic in the remaining steps. If $d \geq p$, degrees of freedom in W_j to W_{k-1} is sufficient, thus only the single choice of IV_{second} is sufficient to find a collision. This is advantageous because the cost of computing the first block directly impacts to the overall attack complexity. If $d < p$, degrees of freedom in W_j to W_{k-1} is insufficient, thus the generation of IV_{second} in the first block must be repeated multiple times.

Whether the attack can be faster than the generic attack depends on the relationship between d and p. To evaluate the attack complexity, we first discuss the complexity in the classical setting. Let 2^f be the complexity to generate an

[11] In Sections 5 and 6, we considered the special case where s is the number of the starting step of a local collision.

IV_{second}.[12] In the classical setting, when $d \geq p$, the attacker needs to generate a single choice of IV_{second} and examine 2^p choices of message words for the last part of the characteristic. Hence, the attack complexity is $2^f + 2^p$. when $d < p$, the attacker needs to generate 2^{p-d} choices of IV_{second} and, for each of them, examine 2^d choices of message words for the last part of the characteristic. Hence, the attack complexity is $2^f \cdot 2^{p-d} + 2^{p-d} \cdot 2^d$, which is equal to $2^f \cdot 2^{p-d} + 2^p$. To be a valid attack, this complexity must be faster than the generic attack complexity, which is $2^{n/2}$ in the classical setting. Hence, $\max(f, p) < n/2$ when $d \geq p$ and $\max(p, f + p - d) \leq n/2$ when $d < p$. The closed formula for both cases is $\max(p, f + \max(p - d, 0)) < n/2$. Therefore, $p < n/2$ must be satisfied in the classical setting, namely, the probability of the differential characteristic for the last part cannot be smaller than $2^{-n/2}$.

The complexity evaluation in the quantum setting is as follows. When $d \geq p$, the cost for generating an IV_{second} and examining 2^p choices of message for the last part of the characteristic decreases to $\sqrt{2^f}$ and $\sqrt{2^p}$, respectively, by using the Grover search. Hence the attack complexity becomes $\sqrt{2^f}+\sqrt{2^p}$. When $d < p$, similarly we obtain quadratic speed up for each subroutine with the Grover search, and the attack complexity decreases to $\sqrt{2^f \cdot 2^{p-d}} + \sqrt{2^p}$. A quantum attack can be valid in the cost metric of time-space tradeoff if its complexity is below $2^{n/2}$, i.e., $\max(p, f + \max(p - d, 0)) < n$. In particular, a quantum attack can be valid even if $p \geq n/2$, i.e., the probability of the differential characteristic for the last part can be smaller than $2^{-n/2}$.

Note that $d \geq p$ may occur in practice. In fact, the 31-step (not 38-step) semi-free-start collision attack by Mendel et al. against SHA-256 is exactly the case with $d \geq p$. As introduced in Sect. 3.1, the authors of [32] explained that Step III succeeds with a probability about 1/12 due to the lack of degrees of freedom in W_{13}, W_{14}, W_{15}. However, if it is analyzed carefully, 1/12 is a part of probability that the generated IV_{second} is suitable, i.e. there are additional condition of 3.5 bits besides the match of the most significant 96 bits. The sparse characteristic in Fig. 3 corresponds to the characteristic from Step 13 to 31 in Table 2. There are 28 conditions on ΔE_{13} to ΔE_{16}, thus $p = 28$. [13] Degrees of freedom exist in all bits of W_{13} to W_{15}, thus $d = 96$. Hence, this is the case with $d \geq p$.

Remark 14. Roughly speaking, the attack of Section 5 (resp., Section 6) is the case with $s = 7$, $j = 22$, $d = 0$, $p > 20$, and $f = 32$ (resp., $s = 8$, $j = 23$, $d = 0$, $p > 13.6$, and $f = 0$).

Suitable choice of s. The step index s is the border between the first and the second part of the characteristic in Fig. 3, where the state value is fully fixed after Step s. Suppose that the length of each message word is w-bit and the internal state size of hash functions is $t \cdot w$-bit. (In the case of SHA-256, $w = 32$

[12] In other words, 2^f is the complexity to find a first block message M that can be connected to FIX_{start}.

[13] While Table 2 shows only 26 conditions on ΔE_{13} to ΔE_{16}, the original paper implies two additional conditions. Hence we deduce that $p = 28$. See also Remark 3.

and $t = 8$.) Then the parameter f increases much and the attack may not work if s is too small or too large compared to t. The reason is as follows.

If s is too small compared to t (e.g., $s < t/2$), then degrees of freedom in $W_0, \ldots, W_{s-1}$ become too small and the probability that a randomly chosen IV_{second} can be connected to FIX_{start} becomes too small. Hence the parameter f becomes too large.

If s is too large (e.g., $s > 2t$), then the degrees of freedom in $W_0, \ldots, W_{s-1}$ will remain enough. However, this time it may be unclear which choice of $W_0, \ldots, W_{s-1}$ is compatible with IV_{second} and FIX_{start} for a random given IV_{second}, and the complexity to find a compatible choice becomes high. This implies that the parameter f also increases in this case.

Therefore we expect that an index s that is close to t (e.g., $t/2 < s < 2t$) will be suitable. (Indeed s is close to t in our attacks in Sections 5 and 6.)

Remark on Memory. Memory is quite expensive in the quantum setting while cheap in the classical setting. [14] Thus differential characteristics that lead to memory-less attacks but seem non-optimal in the classical setting are worth investigating in the quantum setting.

7.2 Towards Application to Other Hash Functions

A natural question that arises after seeing our results is whether we can apply the same idea to other hash functions by using similar differential characteristics shown in previous works. In earlier sections we focused on semi-free-start collisions on SHA-256 and SHA-512, which are single-branch hash functions, but we do not have to restrict ourselves to semi-free-start collisions nor single-branch hash functions: Differential characteristics for free-start collisions or double-branch hash functions may also lead to quantum collision attacks if their structures are close to Fig. 3. [15]

Indeed, some previous works use such differential characteristics. Examples are the semi-free-start collision attack on reduced HAS-160 in [29], the free-start collision attacks on reduced SHA-2 family in [10] and reduced SM3 in [31], and the semi-free-start collision attacks on full RIPEMD-128 in [23] and reduced RIPEMD-160 in [27,33,28]. The differential characteristics used in these attacks look similar to Fig. 3 and they are found by using automated search tools in a similar way to the differential characteristics that we used in Sections 5 and 6.

We investigated whether we could use those differential characteristics to mount quantum 2-block collision attacks. We elaborate observations that we obtained so far in Section G in the appendix. Unfortunately, we have not succeeded

[14] The situation may change if we adopt the cost-metric that assumes the existence of quantum RAM instead of the cost-metric of time-memory tradeoff, but we expect that finding attacks that are valid in the latter is easier than finding ones valid in the former.

[15] Recall that a collision $((\text{IV}, M), (\text{IV}', M'))$ for a compression function h is called a semi-free-start collision if $\text{IV} = \text{IV}'$ and *free-start collision* if $\text{IV} \neq \text{IV}'$.

yet, and with this respect, the analysis here is a failure report. Nevertheless, we believe that those are valuable to report because we observe that some of the applications are close to be valid collision attacks while others are very far. By sharing the experience of those analysis, it would be possible to search for new differential characteristics that satisfy properties in Section 7.1 in order to break more rounds than classical collision attack.

8 Concluding Remarks

In this paper, we showed collision attacks on 38 and 39 steps of SHA-256 and SHA-512, respectively, when the attacker can access to quantum machines under the time-space tradeoff metric. The complexity is $2^{122}/\sqrt{S}$ and $2^{252.7}/\sqrt{S}$ where $S < 2^{12}$ and $S < 2^{6.6}$ for SHA-256 and SHA-512, respectively.

Both attacks followed the same approach as the previous work, where a semi-free-start collision attack that works for 2^X choices of IVs ($X > \frac{n}{2}$) is converted into a 2-block collision. We observed that even a small X may lead to an attack faster than the generic one.

A possible future direction is to study applications to other cryptographic hash functions. Since the idea behind our quantum collision attacks is very simple, we believe that it has broad applications. It will also be interesting to study optimizations of differential characteristics for the classical semi-free-start collision attack with respect to the conversion to the quantum collision attack.

Acknowledgments

We thank anonymous reviewers for their insightful comments, especially for pointing out errors in previous versions of the paper.

References

1. Telecommunications Technology Association. Hash Function Standard Part 2: Hash Function Algorithm Standard (HAS-160), TTAS.KO-12.0011/R1 (2008)
2. ISO/IEC 10118-3, IT Security techniques — Hash-functions — Part 3: Dedicated hash-functions (2018)
3. Aoki, K., Guo, J., Matusiewicz, K., Sasaki, Y., Wang, L.: Preimages for step-reduced SHA-2. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 578–597. Springer (2009)
4. Bernstein, D.J.: Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? In: SHARCS (2009)
5. Biryukov, A., Lamberger, M., Mendel, F., Nikolic, I.: Second-order differential collisions for reduced SHA-256. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 270–287. Springer (2011)
6. Boyer, M., Brassard, G., Høyer, P., Tapp, A.: Tight bounds on quantum searching. Fortschritte der Physik: Progress of Physics **46**(4-5), 493–505 (1998)
7. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: LATIN 1998. LNCS, vol. 1380, pp. 163–169. Springer (1998)

8. Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An efficient quantum collision search algorithm and implications on symmetric cryptography. In: ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 211–240. Springer (2017)
9. Dobbertin, H., Bosselaers, A., Preneel, B.: RIPEMD-160: A strengthened version of RIPEMD. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 71–82. Springer (1996)
10. Dobraunig, C., Eichlseder, M., Mendel, F.: Analysis of SHA-512/224 and SHA-512/256. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 612–630. Springer (2015)
11. Dobraunig, C., Eichlseder, M., Mendel, F.: Analysis of SHA-512/224 and SHA-512/256. IACR Cryptology ePrint Archive 2016/374 (2016), The full version of [10].
12. Dong, X., Sun, S., Shi, D., Gao, F., Wang, X., Hu, L.: Quantum collision attacks on aes-like hashing with low quantum random access memories. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 727–757. Springer (2020)
13. Eichlseder, M., Mendel, F., Schläffer, M.: Branching heuristics in differential collision search with applications to SHA-512. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 473–488. Springer (2014)
14. Flórez-Gutiérrez, A., Leurent, G., Naya-Plasencia, M., Perrin, L., Schrottenloher, A., Sibleyras, F.: New results on gimli: Full-permutation distinguishers and improved collisions. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 33–63. Springer
15. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: ACM STOC 1996. pp. 212–219. ACM (1996)
16. Guo, J., Ling, S., Rechberger, C., Wang, H.: Advanced meet-in-the-middle preimage attacks: First results on full tiger, and improved results on MD4 and SHA-2. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 56–75. Springer (2010)
17. Hosoyamada, A., Sasaki, Y.: Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 249–279. Springer (2020)
18. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-meyer and merkle-damgård constructions. In: Peyrin, T., Galbraith, S.D. (eds.) Asiacrypt 2018, Part I. LNCS, vol. 11272, pp. 275–304. Springer (2018)
19. Indesteege, S., Mendel, F., Preneel, B., Rechberger, C.: Collisions and other nonrandom properties for step-reduced SHA-256. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 276–293. Springer (2008)
20. Isobe, T., Shibutani, K.: Preimage attacks on reduced tiger and SHA-2. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 139–155. Springer (2009)
21. Jaques, S., Schanck, J.M.: Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 32–61. Springer (2019)
22. Khovratovich, D., Rechberger, C., Savelieva, A.: Bicliques for preimages: Attacks on skein-512 and the SHA-2 family. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 244–263. Springer (2012)
23. Landelle, F., Peyrin, T.: Cryptanalysis of full RIPEMD-128. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 228–244. Springer (2013)

24. Leurent, G., Peyrin, T.: SHA-1 is a shambles: First chosen-prefix collision on SHA-1 and application to the PGP web of trust. In: Capkun, S., Roesner, F. (eds.) USENIX Security 2020. pp. 1839–1856. USENIX Association (2020)
25. Li, J., Isobe, T., Shibutani, K.: Converting Meet-In-The-Middle Preimage Attack into Pseudo Collision Attack: Application to SHA-2. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 264–286. Springer (2012)
26. Liu, F., Dobraunig, C., Mendel, F., Isobe, T., Wang, G., Cao, Z.: Efficient collision attack frameworks for RIPEMD-160. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 117–149. Springer (2019)
27. Liu, F., Dobraunig, C., Mendel, F., Isobe, T., Wang, G., Cao, Z.: New semi-free-start collision attack framework for reduced RIPEMD-160. IACR Trans. Symmetric Cryptol. **2019**(3), 169–192 (2019)
28. Liu, F., Mendel, F., Wang, G.: Collisions and semi-free-start collisions for round-reduced RIPEMD-160. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 158–186. Springer (2017)
29. Mendel, F., Nad, T., Schläffer, M.: Cryptanalysis of round-reduced HAS-160. In: Kim, H. (ed.) ICISC 2011. LNCS, vol. 7259, pp. 33–47. Springer (2011)
30. Mendel, F., Nad, T., Schläffer, M.: Finding SHA-2 characteristics: Searching through a minefield of contradictions. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 288–307. Springer (2011)
31. Mendel, F., Nad, T., Schläffer, M.: Finding collisions for round-reduced SM3. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 174–188. Springer (2013)
32. Mendel, F., Nad, T., Schläffer, M.: Improving local collisions: New attacks on reduced SHA-256. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 262–278. Springer (2013)
33. Mendel, F., Peyrin, T., Schläffer, M., Wang, L., Wu, S.: Improved cryptanalysis of reduced RIPEMD-160. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 484–503. Springer (2013)
34. Mendel, F., Pramstaller, N., Rechberger, C., Rijmen, V.: Analysis of step-reduced SHA-256. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 126–143. Springer (2006)
35. Mendel, F., Rijmen, V.: Colliding message pair for 53-step HAS-160. In: Nam, K., Rhee, G. (eds.) ICISC 2007. LNCS, vol. 4817, pp. 324–334. Springer (2007)
36. Mitzenmacher, M., Upfal, E.: Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis (2nd edition). Cambridge university press (2017)
37. National Institute of Standards and Technology: Secure Hash Standard (SHS), FIPS PUB 180-4 edn. (August 2015)
38. Nikolić, I., Biryukov, A.: Collisions for step-reduced SHA-256. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 1–15. Springer (2008)
39. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with application to hash functions and discrete logarithms. In: ACM CCS 1994. pp. 210–218. ACM (1994)
40. Sanadhya, S.K., Sarkar, P.: 22-step collisions for SHA-2. CoRR **abs/0803.1220** (2008)
41. Sanadhya, S.K., Sarkar, P.: New collision attacks against up to 24-step SHA-2. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 91–103. Springer (2008)
42. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., Markov, Y.: The first collision for full SHA-1. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 570–596. Springer (2017)

43. Wang, G.: Practical collision attack on 40-step RIPEMD-128. In: Benaloh, J. (ed.) CT-RSA 2014. LNCS, vol. 8366, pp. 444–460. Springer (2014)
44. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Info. Comput. **15**(7-8), 557–567 (May 2015)

A Collisions, Semi-Free-Start Collisions, and Free-Start Collisions

This section describes the difference among collisions, semi-free-start collisions, and free-start collisions of a hash function that is based on the Merkle-Damgård construction. Recall that the Merkle-Damgård construction converts a compression function $h : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ of fixed input-output length into a hash function $\mathcal{H}_{\mathrm{IV}_0} : \{0,1\}^* \to \{0,1\}^n$ of variable input lengths as follows.

1. Pad an input message M so that its length will be a multiple of m, by using a padding function with a favorable property. Let $\bar{M} = M[1]||\cdots||M[\ell]$ be the message after padding.
2. Set the first state to be $S_0 := \mathrm{IV}_0$. (Usually IV_0 is a certain fixed constant specified by the designers of the hash function.)
3. Update the states as $S_i := h(M[i], S_{i-1})$ for $i = 1, \ldots, \ell$.
4. Output S_ℓ (i.e., $\mathcal{H}_{\mathrm{IV}_0}(M) := S_\ell$).

A *collision* of the hash function $\mathcal{H}_{\mathrm{IV}_0}$ is a pair of distinct messages M, M' such that $\mathcal{H}_{\mathrm{IV}_0}(M) = \mathcal{H}_{\mathrm{IV}_0}(M)$. On the other hand, a *semi-free-start collision* of the hash function $\mathcal{H}_{\mathrm{IV}_0}$ is a pair of distinct messages and initial values $(M, \mathrm{IV}), (M'\mathrm{IV}')$ such that $\mathcal{H}_{\mathrm{IV}}(M) = \mathcal{H}_{\mathrm{IV}'}(M)$ and $\mathrm{IV} = \mathrm{IV}'$ hold, but IV *is not necessarily equal to the originally specified initial value* IV_0. If IV is not equal to IV′, the pair $((M, \mathrm{IV}), (M', \mathrm{IV}'))$ is called a *free-start-collision*.

The notion of ((semi-)free-start) collisions are defined for the compression function h in the same way. Note that a ((semi-)free-start) collision of h can be trivially extended into a ((semi-)free-start) collision of $\mathcal{H}$ if each message M is a prefix of the padded message $\bar{M}$, which is the case for usual concrete hash functions.

B Discussions on the Generic Collision Attack based on the Multi-Target Preimage Search

This section provides comparisons of our dedicated attacks with the generic collision attack based on the multi-target preimage search.

When a quantum computer of size $O(S)$ is available, we can mount a simple collision attack based on the multi-target preimage search with time complexity $O(\sqrt{2^n/S})$. The complexity $O(\sqrt{2^n/S})$ is of course asymptotically worse than the generic complexity $O(\sqrt{2^n}/S)$ in the setting of time-space tradeoff. However, some reader may wonder if the *concrete* complexity of the former becomes lower than the latter, or even the time complexity of our dedicated attacks on reduced SHA-256 and SHA-512 (and thus our attacks are invalid).

In what follows we show we do not have to worry about this, i.e., the complexity of our dedicated attacks are lower than that of the multi-target preimage search.

Collision Attack based on Multi-Target Preimage Search. First we describe an overview of the collision attack based on multi-target preimage search. Suppose that a quantum computer of size $O(S)$ is available (we assume $S \ll 2^{n/3}$). Then, we can find a collision of a hash function H of n-bit output in time $O\left(\sqrt{2^n/S}\right)$ as follows.

1. Compute the values $y_1 = H(x_1), \ldots, y_S = H(x_S)$ for randomly chosen elements $x_1, x_2, \ldots, x_S$ and store them in a list L.
2. By using the Grover search, find x' such that $x' \neq x_1, \ldots, x_S$ and $H(x')$ is in L.

The Grover search of Step 2 finds a desired x' in time $\frac{\pi}{4}\sqrt{2^n/|L|} = O(\sqrt{2^n/S})$. Since Step 1 runs in time $O(S)$ and $S \leq \sqrt{2^n/S}$ if $S \ll 2^{n/3}$, the above attack runs in time $O(\sqrt{2^n/S})$. In what follows we denote this collision attack by MTPS for short.

Comparison with Our Dedicated Attacks. For instance, suppose $n = 256$ and about 3000 qubits are available as in our attack on reduced SHA-256 (without parallelization). Since $3000/256 \approx 11$, we assume that $S = 11$ values are stored into L in the first step of MTPS. In each Grover iteration of the second step of MTPS, for a given x', we have to evaluate $H(x')$ twice to examine whether $H(x') \in L$ because uncomputations are required. Thus the *concrete* time complexity of MTPS in this setting is $\frac{\pi}{4} \cdot 2 \cdot \sqrt{2^{256}/11} \approx 2^{127}$. Since the time complexity of our attack with 3000 qubits is 2^{122}, the margin compared to 2^{127} is large enough. Similar arguments apply for our dedicated attack on reduced SHA-512.

C Details on the Success Probability of the Parallelized Grover Search

This section shows that we can find x such that $F(x) = 1$ with probability at least $1 - 1/e$ if we run P copies of $\mathsf{Grov}(F, \lfloor \pi/4\theta\sqrt{P} \rfloor)$ (the Grover search with $X := \lfloor \pi/4\theta\sqrt{P} \rfloor$ iterations). Recall that we assume $(|F^{-1}(1)| \cdot P)/2^n \ll 1$, which implies that $\theta \cdot P \ll 1$.

First, a single machine finds a solution with probability $\sin^2((2X+1)\theta)$. Thus, the probability that at least one machine (among P machines) finds a solution is

$$p_{\mathsf{succ}} := 1 - (1 - \sin^2((2X+1)\theta))^P.$$

Since $\sin x \geq x - x^3/6$ holds for $x \geq 0$,

$$1 - (1 - \sin^2((2X+1)\theta))^P \geq 1 - \left(1 - \left((2X+1)\theta - \frac{((2X+1)\theta)^3}{6}\right)^2\right)^P$$

holds. In addition, since $1 - x \leq e^{-x}$ holds,

$$1 - \left(1 - \left((2X+1)\theta - \frac{((2X+1)\theta)^3}{6}\right)^2\right)^P \geq 1 - \left(e^{-\left((2X+1)\theta - \frac{((2X+1)\theta)^3}{6}\right)^2}\right)^P$$

follows. Because $\sqrt{2} > (2X+1)\theta \geq \pi/2\sqrt{P} - \theta > 0$ holds when $P \geq 2$ and $P \cdot \theta \ll 1$, and $x - x^3/6$ decreases as x decreases for $0 < x < \sqrt{2}$,

$$\begin{aligned}1 - \left(e^{-\left((2X+1)\theta - \frac{((2X+1)\theta)^3}{6}\right)^2}\right)^P &\geq 1 - \left(e^{-\left(\left(\pi/2\sqrt{P} - \theta\right) - \frac{\left(\pi/2\sqrt{P} - \theta\right)^3}{6}\right)^2}\right)^P \\ &= 1 - e^{-\left(\frac{\pi}{2} - \sqrt{P}\theta\right)^2 \left(1 - \frac{\left(\pi/2\sqrt{P} - \theta\right)^2}{6}\right)^2}\end{aligned}$$

holds. Moreover, since $\left(\frac{\pi}{2} - \sqrt{P}\theta\right)^2 \left(1 - \frac{\left(\pi/2\sqrt{P} - \theta\right)^2}{6}\right)^2 \geq 1$ holds when $P \geq 2$ and $P \cdot \theta \ll 1$, we have

$$1 - e^{-\left(\frac{\pi}{2} - \sqrt{P}\theta\right)^2 \left(1 - \frac{\left(\pi/2\sqrt{P} - \theta\right)^2}{6}\right)^2} \geq 1 - 1/e.$$

Therefore $p_{\mathsf{succ}} \geq 1 - 1/e$ follows.

D Details on Remark 11

Some readers may consider that our complexity analysis in Section 5 is invalid since, for a function $F : \{0,1\}^n \to \{0,1\}$, the complexity $(\pi/4)\sqrt{2^n/|F^{-1}(1)|}$ of the Grover search can be achieved only when $|F^{-1}(1)|$ is known in advance (or, equivalently, the exact value of $p = |F^{-1}(1)|/2^n$ is known in advance). However, we can reasonably expect that p is so close to $p_0 := 2^{-32} \cdot \left(1179647/(2^{32})^7\right)$, namely, the success probability of the Grover search does not significantly change even if we estimate that p is equal to p_0 and compute the number of Grover's iterations from p_0 but not from p.

This section explains the reason that the Grover search succeeds with a high probability and the analysis in Section 5 works even if we estimate that p is equal to $p_0 := 2^{-32} \cdot \left(1179647/(2^{32})^7\right)$, i.e., we estimate p under the assumption that $\mathrm{IV}_{\mathrm{second}}$ is chosen uniformly at random, and compute the number of Grover's iteration from p_0 but not from p.

More precisely, let θ (resp., θ_0) denote the value such that $0 \leq \theta \leq \pi/2$ and $\sin^2\theta = p$ (resp., $0 \leq \theta_0 \leq \pi/2$ and $\sin^2\theta_0 = p_0$). In addition, let $m := \lfloor \pi/4\theta \rfloor$ and $m_0 := \lfloor \pi/4\theta_0 \rfloor$. Estimating p to be p_0 means estimating m to be m_0. As mentioned in Section 2, what Boyer et al. showed is that $\mathsf{Grov}(F, m)$ outputs x such that $F(x) = 1$ with an overwhelming probability. What we show below is that $\mathsf{Grov}(F, m_0)$ also outputs x such that $F(x) = 1$ with an overwhelming probability.

Following usual classical cryptanalysis, we heuristically assume that the compression function of the first block $f(\mathrm{IV}, \cdot)$ will behave like a random function $g : \{0,1\}^{512} \to \{0,1\}^{256}$. Let $S \subset \{0,1\}^{256}$ be the set of $\mathrm{IV}_{\mathrm{second}}$ such that both of the conditions in Steps II-(a) and II-(b) are satisfied ($|S| = 1179647$ holds). Then $|g^{-1}(S)|$ is a random variable that follows the binomial distribution such that the success probability of each trial is $p_0 = |S|/2^{256}$ and the number of trials is $N := 2^{512}$. In addition, $p = |g^{-1}(S)|/N$ holds.

Let $\mu_0 := p_0 \cdot N = |S| \cdot \sqrt{N}$. From the Chernoff bound,

$$\Pr\left[\left|\mu_0 - \left|g^{-1}(S)\right|\right| \le \delta \cdot \mu_0\right] \ge 1 - 2e^{-\mu_0 \delta^2/3}$$

holds for arbitrary δ such that $0 < \delta < 1$ (see, for instance, Corollary 4.6 of [36]). By setting $\delta := \sqrt{30/\mu_0}$ we have

$$\Pr\left[\left|\mu_0 - \left|g^{-1}(S)\right|\right| \le \sqrt{30\mu_0}\right] \ge 1 - 2e^{-10}. \tag{5}$$

Since $p = |g^{-1}(S)|/N$ and $p_0 = \mu_0/N$ hold, the above inequality is equivalent to

$$\Pr\left[|p_0 - p| \le \sqrt{\frac{30p_0}{N}}\right] \ge 1 - 2e^{-10}. \tag{6}$$

Since $\sqrt{30p_0/N} = \sqrt{p_0} \cdot \sqrt{30/N} < p_0^{3/2}$ holds, from (6) it follows that

$$(1 - \sqrt{p_0}) \cdot p_0 \le p \le (1 + \sqrt{p_0}) \cdot p_0 \tag{7}$$

holds with an overwhelming probability, which is equivalent to

$$\sqrt{1 - \sqrt{p_0}} \cdot \sin\theta_0 \le \sin\theta \le \sqrt{1 + \sqrt{p_0}} \cdot \sin\theta_0. \tag{8}$$

Since $\theta_0 \ll 1$, we can expect that

$$(1 - \epsilon) \cdot \theta_0 \le \theta \le (1 + \epsilon) \cdot \theta_0 \tag{9}$$

holds with an overwhelming probability, where $\epsilon \approx \max\left\{1 - \sqrt{1 - \sqrt{p_0}},\ \sqrt{1 + \sqrt{p_0}} - 1\right\}$.

Recall that the success probability of $\mathsf{Grov}(F, m)$ and $\mathsf{Grov}(F, m_0)$ are $\sin^2((2m+1)\theta)$ and $\sin^2((2m_0+1)\theta_0)$, respectively, where $m = \lfloor 4/\pi\theta \rfloor$ and $m_0 = \lfloor 4/\pi\theta_0 \rfloor$. From (9) it follows that $\frac{1}{1+\epsilon} \cdot m_0 - 1 \le m \le \frac{1}{1-\epsilon} \cdot (m_0 + 1)$ holds, which implies that

$$|m - m_0| \le \frac{\epsilon}{1-\epsilon} m_0 + \frac{1}{1-\epsilon} \le \frac{\epsilon(1+\epsilon)}{1-\epsilon}(m+1) + \frac{1}{1-\epsilon} \le 2\epsilon m + 2 \tag{10}$$

holds (since $\epsilon \ll 1$). Since $|\frac{\pi}{2} - ((2m+1)\theta)| \le \theta$ holds by definition of m,

$$\begin{aligned}
\left|\frac{\pi}{2} - ((2m_0+1)\theta)\right| &\le \left|\frac{\pi}{2} - ((2m+1)\theta)\right| + 2|m - m_0|\theta \\
&\le \theta + 2|m - m_0|\theta \\
&\le \theta + 2 \cdot (2\epsilon m + 2)\theta \\
&\le 5\theta + 2\epsilon \cdot (2m+1)\theta \\
&\le 2\epsilon \cdot \left(\frac{\pi}{2} + \epsilon\right) + 5\theta
\end{aligned}$$

follows. Since $\theta, \epsilon \ll 1$ holds, we can expect that $\Pr[\mathsf{Grov}(F, m_0) \text{ succeeds}] = \sin^2((2m_0+1)\theta)$ is sufficiently close to 1.

E Additional Figures

Table 5. The 39-step differential characteristic for SHA-512 shown in [11].

i	ΔA_i	ΔE_i	ΔW_i
−4	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	
−3	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	
−2	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	
−1	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	
0	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
1	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
2	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
3	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
4	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
5	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
6	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- -10- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
7	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- -01- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
8	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
9	---u uuu- u--- u--- ---- ---u uuuu u--- -unn ---- ---- ---- ---- -unn n-nn ----	nnn1 -01n n11u n1-- ---- -0-- 00-- 101n uuun 00-- --00 0--- 0-1- ---- -u0u 1---	--u- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---n ----
10	nun- uuu- u-u- u-nu ---- -uun --n- --n- ---- -uuu -n-n u--- n-u- n--n un-u u-n-	101- --0n 0011 1000 n1n- 0nu- 00-- 1n11 1u0u n-10 0-nn n--0 u-0u 10-1 10u- 1-0-	-1-- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- --u- ----
11	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	n11- --1u 1un0 0uu1 000n 0u00 10-- 1nu0 11u1 01-0 0n01 n1-0 1-n- 1nn0 011n 1u10	---- ---- -nnn nnn- ---- -n-- ---- ---- ---- ---- ---- ---- --nu uuuu u--- ---n
12	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	1101 nnn0 n111 n10u 0100 0100 nn1n n1u0 1101 n11u 1011 0-01 0110 1n1u 10n1 n000	--nu uuuu uuuu uu-- ---- ---- ---- ---- ---- ---- ---- ---- ---- --1u 0--1 111-
13	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	10-1 10n1 u00n n00n 1000 1111 0100 n111 0001 uu11 n1nn 0111 1011 1uuu uu0u 1011	---- --1- ---- ---- ---- ---- ---- ---- -1-- ---- ---- ---- ---- ---- ---- ----
14	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	1n1n 10n0 uu0n n00n 0001 0u0u 01n0 01n0 0000 u10n 1n01 n101 n1un u01n 0uu0 nnu0	---- ---- ---- ---- ---- ---- -1-- ---- ---- ---- --1- ---- ---- ---- ---1 ----
15	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	-0-1 ---- -1-- ---- ---- -011 --1- 011- ---- -101 1011 0--- 0100 0--- 1001 000-	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
16	---- ---- --u- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- u---	11-1 --1- --u- --10 ---- -100 --1- -00- -11- -100 00-- 0--0 1001 0001 1unn n11-	--u- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---n ----
17	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	1111 1100 111- 101- -0-- 00un n101 0nu0 11nu nuu0 00-- u--0 0u00 0001 1110 0011	-1-- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- --u- ----
18	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- --u- ----	0111 1000 00n- 00-- 01-- 001- ---- 1110 1unn nnnn nnnn nnnn nnnn nnnn u100	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
19	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	uuuu uuuu uuu- nuuu uuuu uuu0 01-0 uunn nn01 0000 1100 010n uuuu uuuu uuu0 n01-	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
20	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- 1011 11u0 --10 0000 000- ---- 0--0 11-- -101 1111 0110 1101 nuuu uuu1 u---	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
21	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	1111 1111 11-- 1111 1111 111- ---- 1111 11-- -101 1111 0110 0000 0111 110- 1---	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
22	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	0-0- 0-0- 0-1- 1-0- 0-1- ---- ---- ---0 -0un nnnn nnnn nnnn nnnn nnn- 0-0-	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
23	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- --11 1111 1111 1111 1111 1111 111- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
24	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- --11 1111 1111 1111 1111 1111 111- ----	---- ---- --n- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- n---
25	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
26	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---n ----
27	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
28	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
29	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
30	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
31	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
32	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
33	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
34	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
35	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
36	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
37	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
38	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----	---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----

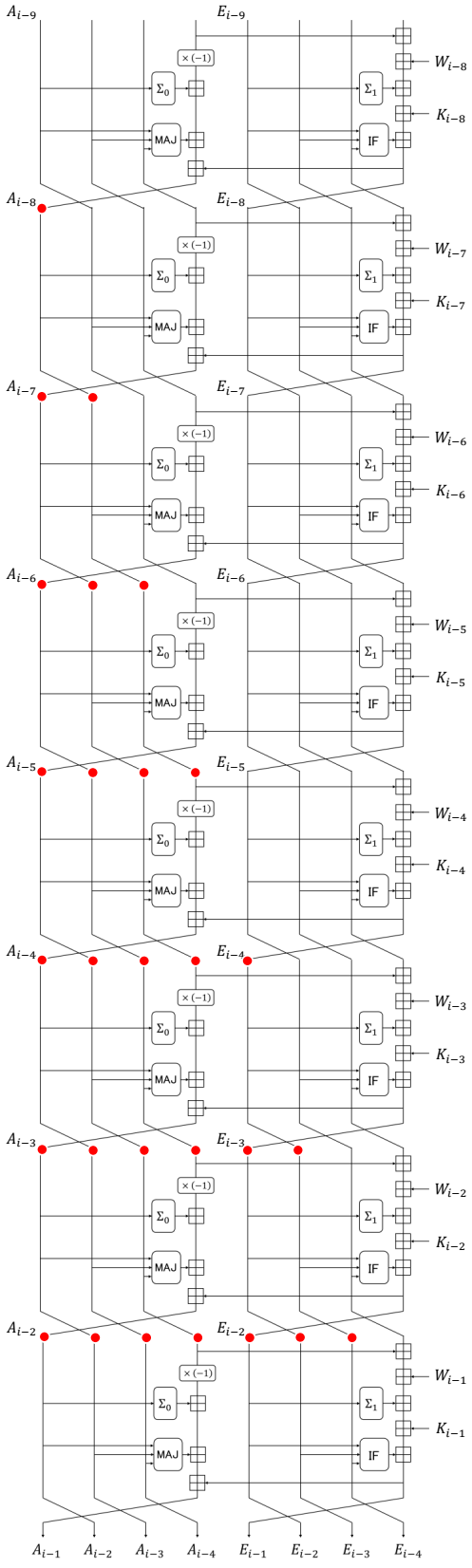

Fig. 4. The internal state variables (pointed with red dots) that are determined by computing step functions backwards from the internal state $A_{i-1}||\cdots||A_{i-4}||E_{i-1}||\ldots||E_{i-4}$ at the beginning of step i.

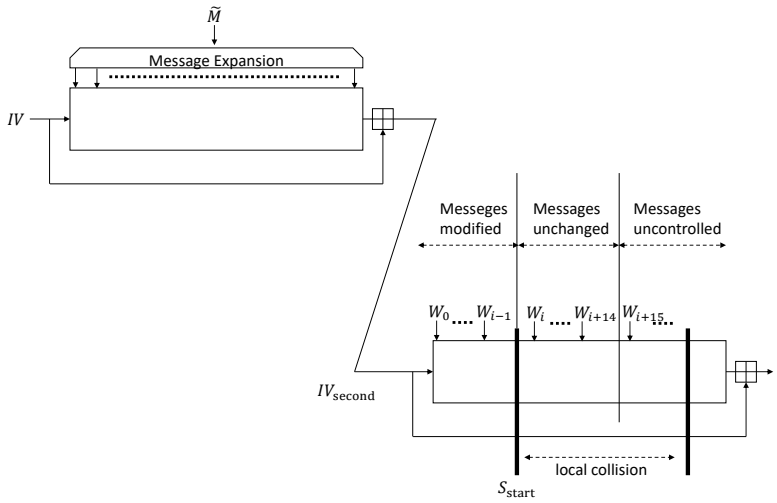

Fig. 5. The idea of our quantum attack.

F (Semi-free-start) Collision Examples from Previous Work

Table 6. An example of a semi-free-start collision of 31-step SHA-256 shown in [32]. $\Delta M := M \oplus M'$. h_0 is an initial value of the compression function. The last 5 words of h_0 are set to be 0 to show the first 5 message words can be chosen freely.

h_0	532f13f5 6a28c3c0 e301fab5 00000000 00000000 00000000 00000000 00000000
M	d55c884f faf18f34 b772b323 af46235b 3d8bd87b dd3e8271 26618488 02d189d0
	1883a4af 4f99167b 271b11c7 81b8363d b27e389d 2155a533 8b811348 4a8da291
M'	d55c884f faf18f34 b772b323 af46235b 3d8bd87b 523f9273 eeb902ae 36ff3d98
	108477b0 4f989677 271b11c7 81b8363d b27e389d 2155a533 8b811348 4a8da291
ΔM	00000000 00000000 00000000 00000000 00000000 8f011002 c8d88626 342eb448
	0807d31f 0001800c 00000000 00000000 00000000 00000000 00000000 00000000

Table 7. The semi-free-start collision of 38-step SHA-256 shown in [32].

h_0	ba75b4ac c3c9fd45 fce04f3a 6d620fdb 42559d01 b0a0cd10 729ca9bc b284a572
M	4f5267f8 8f8ec13b 22371c61 56836f2b 459501d1 8078899e 98947e61 4015ef31
	06e98ffc 4babda4a 27809447 3bf9f3be 7b3b74e1 065f711d 6c6ead5e a1781d54
M'	4f5267f8 8f8ec13b 22371c61 56836f2b 459501d1 8078899e 98947e61 7e73f1f1
	06e99000 4babda4a 277f1447 3bf9f3be 7b3b74e1 065f711d 6c6ead5e a1781d50
ΔM	00000000 00000000 00000000 00000000 00000000 00000000 00000000 3e661ec0
	00001ffc 00000000 00ff8000 00000000 00000000 00000000 00000000 00000004

Table 8. The semi-free-start collision of 39-step SHA-512 shown in [10].

h_0	eccf3da189dd9668	b1ec21a4fd53b8d8	609ce4465f772770	adf4e7738e2978f6
	8edd237ea50eebc9	231b3af0102a926d	db45e613e8d2fd52	ad384433420073f6
M	a0ec9872cfffe63c	df5c6a2b59f4c453	f2bea3763fc8fa7a	6a47e8ff0a995116
	fa59232e8b617048	4c9690984c084498	28bee8f5701eab16	8d57686ecbdce623
	3879318f901ff782	72644b0ca55a6142	6cb281dab11480b4	4a8198441f401ff2
	5ffd956ed11a2b5f	9a640988d68287d3	74942df792f2637f	b2819dc61f772d4f
M'	a0ec9872cfffe63c	df5c6a2b59f4c453	f2bea3763fc8fa7a	6a47e8ff0a995116
	fa59232e8b617048	4c9690984c084498	28bee8f5701eab16	8d57686ecbdce623
	3879318f901ff7a2	52644b0ca55a6152	6cb281dab1148094	4aff9c441f402073
	6001956ed11a2a5f	9a640988d68287d3	74942df792f2637f	b2819dc61f772d4f
ΔM	0000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000000	0000000000000000	0000000000000000	0000000000000000
	0000000000000020	2000000000000010	0000000000000020	007e040000003f81
	3ffc000000000100	0000000000000000	0000000000000000	0000000000000000

G Observations on Applications to Other Hash Functions

This section elaborates observations that we obtained so far on applications of our quantum collision attacks to other hash functions.

On Application to HAS-160. HAS-160 is a single-branch hash function with 160 bits output standardized by the Korean government [1]. The best collision attack reaches 53 steps [35] and the best semi-free-start collision attack reaches 65 steps [29].

We could not find a valid quantum collision attack based on the 65-step differential characteristic in [29] because the local collision starts too late and we have $s > 16$ and $t = 5$ (with the notations in Section 7.1).

However, we still expect that there exist a quantum collision attack that breaks more rounds than classical attacks: The conditions in the 65-step differential characteristic is very sparse (i.e., the probability for steps $j, j+1, \ldots$ in Fig. 3 will be high). Thus it may be possible to find a new differential characteristic for 54-65 steps such that its conditions are sparse as well as the previous 65-step characteristic while the local collision starts earlier.

Inapplicability to Free-Start Collisions. Unfortunately, the idea is unlikely to be applicable to free-start collisions in general. This is because the cost to choose the first message block increases.

In the 2-block collision attack based on semi-free-start collisions, first we randomly choose a *single message* M for the first block, and then check whether M leads to a collision at the second block. We have to try many M, but the cost for choosing a single M is negligible.

On the other hand, if we apply the idea to free-start collisions, first we have to randomly choose a *pair of messages* (M, M') for the fist block of which output difference is equal to a specific non-zero value (i.e., the difference $\mathrm{IV} \oplus \mathrm{IV}' \neq 0$ for the initial values of the second block). The cost of choosing such a pair will be equal to that of the generic attack[16], which means that the collision attack is invalid.

On Application to RIPEMD-128/160. RIPEMD-128 and RIPEMD-160 [9] are double-branch hash functions with 128 and 160 bits output, respectively, which are standardized by ISO/IEC [2]. RIPEMD-160 is used by Bitcoin together with SHA-256.

RIPEMD-128. While there already exists a semi-free-start collision attack on full RIPEMD-128 [23], collision attacks reach only up to 40 (out of 64) steps [43]. Hence it is worth studying whether we can mount a quantum collision attack on full RIPEMD-128.

We observed that it is hard to use the full-step differential characteristic in [23] for quantum collision attacks due to the lack of degrees of freedom in message words. Though, it may be possible to mount a quantum collision attack with another differential characteristic that satisfy the properties that we mentioned in Section 7.1.

RIPEMD-160. The current best semi-free-start collision attack on RIPEMD-160 is the one by Liu et al. that reaches 40 (out of 80) steps [27]. The differential characteristic used in the attack has dense conditions in the left branch while sparse conditions in the right branch. The attack searches for semi-free-start collisions for the left branch, expecting that the conditions of the right branch will be satisfied probabilistically.

If we can mount a 2-block quantum collision attack based on the 40-step differential characteristic, it significantly improves the classical collision attack for 34 steps [26]. However, the 40-step characteristic does not seem suitable for quantum 2-block collision attacks. The reason is that the local collision of the left branch starts too late, and we have $s > 16$ and $t = 8$ with the notations in Section 7.1. 36/37/38-step semi-free-start collision attacks are also shown together with the 40-step attack in the same paper but their differential characteristics do not seem suitable for quantum collision attacks either due to the same reason.

The semi-free-start collision attack on 36-step RIPEMD-160 in [33] uses another differential characteristic, which has dense conditions in both branches. The attack first fixes internal state variables and message words for the dense part, and then makes the chaining variables of the both branches match by using degrees of freedom in message words of early steps. The complexity of the attack (with the same differential characteristic) is later improved in [28].

[16] Of course there might be a clever strategy to find such a pair with small cost, but we did not find any.

The 36-step differential characteristic satisfies some properties in Section 7.1, e.g., the index s does not seem too small nor too large, but still we could not find a valid collision attack. It might lead to a quantum collision attack if we had a little bit more degrees of freedom in message words in early steps (i.e., $W_0, \ldots, W_{s-1}$ in Fig. 3) or the probability of the sparse part in later steps (i.e., steps $j, j+1, \ldots,$ in Fig. 3) were slightly higher.

Due to the above observations, we think that it is worthwhile to re-investigate differential characteristics for 35-40 steps of RIPEMD-160, taking into account the possibility of quantum collision attacks and properties described in Section 7.1.