Quantum Attacks on Hash Constructions with Low Quantum Random Access Memory*

Xiaoyang Dong[1,5,6], Shun Li[2], Phuong Pham[2], and Guoyan Zhang[3,4,6]

[1] Institute for Advanced Study, BNRist, Tsinghua University, Beijing, China
xiaoyangdong@tsinghua.edu.cn
[2] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, shun.li@ntu.edu.sg, pham0079@e.ntu.edu.sg
[3] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China, guoyanzhang@sdu.edu.cn
[4] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
[5] Zhongguancun Laboratory, Beijing, China
[6] Shandong Institute of Blockchain, Jinan, China

Abstract. At ASIACRYPT 2022, Benedikt, Fischlin, and Huppert proposed the quantum herding attacks on iterative hash functions for the first time. Their attack needs exponential size of quantum random access memory (qRAM). As the existence of large qRAM is questionable, Benedikt et al. left open question for building low-qRAM quantum herding attacks.

In this paper, we answer this open question by building a quantum herding attack, where the time complexity is slightly increased from Benedikt et al.'s $2^{0.43n}$ to ours $2^{0.46n}$, but the size of qRAM is reduced from Benedikt et al.'s $2^{0.43n}$ to ours $\mathcal{O}(n)$. Besides, we also introduce various low-qRAM quantum attacks on hash concatenation combiner, hash XOR combiner, Hash-Twice, and Zipper hash functions.

Keywords: Quantum computation · qRAM · Herding Attack · Hash Combiner

1 Introduction

Shor's seminal work [51] shows that sufficiently large quantum computers allow factorization of large numbers and computation of discrete logarithms in polynomial time, potentially dooming many public-key schemes in use today. In order to meet the future, the public-key cryptography community and standardization organizations have invested a lot of effort in the research of post-quantum public-key schemes. In particular, NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptography algorithms [48]. In contrast, research on how quantum computing could change the security landscape of symmetric-key cryptography appears to be less active.

* This is a preliminary version that will be revised and updated soon.

For nearly last two decades, it has been generally accepted that Grover's algorithm [30] with quadratic speedup in an exhaustive search attack is the only quantum advantage for symmetric-key cipher, and thus doubling the key length solves this problem.

This view started to change with the initial work of Kuwakado and Morii, who showed that the classically provably secure Even-Mansour cipher and the three-round Feistel network can be broken in polynomial time with the help of quantum computers [40,41]. A few years later, more quantum cryptanalysis of symmetric primitives emerges [36,42,13,34,12,14,22]. Most of these attacks that enjoy exponential speedup rely on Simon's algorithm [52] to find a key-dependent hidden period where access to a quantum superposition oracle of key primitives is necessary. This is a fairly strong claim, and its actual relevance is sometimes questioned. Therefore, a more complex attack still makes sense if it does not require online queries to the superposition oracles of the keyed primitives [10,31,15,11].

For keyless primitives, especially hash functions, quantum attacks are easier to launch, since there is no need for online queries and all computations are public that can be done offline. The classical algorithm finds collisions of n-bit output hash functions with time complexity $\mathcal{O}(2^{n/2})$. In the quantum setting, the BHT algorithm [17] finds collisions with a query complexity of $\mathcal{O}(2^{n/3})$ if $\mathcal{O}(2^{n/3})$ quantum random access memory (qRAM) is available. However, it is generally acknowledged that the difficulty of fabricating large qRAMs is enormous [28,27]. So quantum algorithms (even has relatively high time complexity) using less or no qRAM is desirable. At ASIACRYPT 2017, Chailloux, Naya-Plasencia and Schrottenloher first overcome the $\mathcal{O}(2^{n/2})$ classical bound without using large qRAM [18]. The time complexity of the algorithm is $\mathcal{O}(2^{2n/5})$, the quantum memory is $\mathcal{O}(n)$, and the classical memory is $\mathcal{O}(2^{n/5})$. Also, a quantum algorithm for the generalized birthday problem (or the k-XOR problem) in settings with and without large qRAMs can be found in [29,47]. Besides the generic attacks on hash functions, the first dedicated quantum attack on hash functionss was presented at EUROCRYPT 2020 by Hosoyamada and Sasaki [32], showing quantum attacks on AES-MMO and Whirlpool by exploring differentials whose probability is too low to be useful in the classical setting. Later, refined collision and preimage attacks on hash functions have been presented subsequently by Dong et al. [22,24,23], Flórez Gutiérrez et al. [25], Hosoyamada and Sasaki [33], Schrottenloher and Stevens [50].

The Merkle-Damgård construction [19,46] is a popular way to build hash functions, where a single compression function is iteratively called to extend the input domain from a fixed length to arbitrary length and the digest length is usually the same as that of internal state. However, some widely deployed hash function standards (such as MD5 and SHA-1) based Merkle-Damgård construction have been broken [54,55,53]. Besides, Kelsey and Schneier [38] have demonstrated a generic second-preimage attack against all hash functions based on the classical Merkle-Damgård construction, when the challenge message is long. At CRYPTO 2004, Joux [35] introduced multi-collision attacks on iterated hash functions. At EUROCRYPT 2006, Kelsey and Kohno [37] proposed the herding attack, that the adversary committed to a hash value T of an iterated hash function $\mathcal{H}$, such that when later given a message prefix P, the adversary is able to find a suitable "suffix explanation" S with $\mathcal{H}(P\|S) = T$.

In order to obtain a more secure hash function, and to ensure compatibility, researchers and developers try to combine the two outputs of two (or more) independent hash functions to provide better security in case one or even both hash functions are weak. Practical examples can be found in TLS [20] and SSL [26]. There are several common hash combiners, such as concatenation combiner [49], XOR combiner, Hash-Twice [2], and Zipper hash [44]. However, the security of these hash combiners has also been challenged. At CRYPTO 2004, Joux [35] revealed that the concatenation combiner provides at most $n/2$-bit security for collision resistance and n-bit security for preimage resistance. Leurent and Wang [43] and Dinur [21] showed that the combiners may even weaker than each hash function. Besides, various cryptanalysis results [3,2,45,1,6,4] have been achieved on the hash combiners.

At ASIACRYPT 2022, Benedikt, Fischlin, and Huppert [7] considered quantum nostradamus attacks on iterative hash functions for the first time, and realized attacks of complexity $\mathcal{O}(2^{3n/7})$. The attack requires exponentially large qRAM, which is inherited from the BHT algorithm [17]. Since fabricating large qRAMs is difficult to realize [28,27], Benedikt et al. [7] left open questions for building qRAM-free or low-qRAM quantum herding attack. In 2022, Bao et al. [5] built a low-qRAM quantum herding attack based Chailloux et al.'s multi-target preimage algorithm [18]. However, we find their algorithm is flawed and incorrect when building diamond structure for herding. Therefore, the question is still open.

Our contributions.

In this paper, **for the first contribution,** we answer the open question by Benedikt et al. [7] to build the first valid low-qRAM quantum herding attack on iterated hash functions. We first convert the quantum diamond-building algorithm (it needs exponential large qRAM, i.e., $2^{3n/7}$) proposed by Benedikt et al. into a low-qRAM algorithm that only needs about $2n$ bits of qRAM. The new algorithm is highly based on Chailloux et al.'s collision finding algorithm [18] with various adaptions. In our herding attack, we choose the leaves of the diamond structure to be prefixed with r-bit zeros, then again applying Chailloux et al.'s collision finding to find the linking message S such that $H(P\|S)$ hits one of the leaves of the diamond structure. Note a previous work by Bao et al. [5] also built a quantum herding attack. However, in their attack, the Chailloux et al.'s multi-target preimage algorithm [18] is applied, which can not take the advantage of the ability that attacker can choose the prefixed leaves of the diamond structure. Since Bao et al.'s low-qRAM attack was proved incorrect [5], this paper becomes the first one to propose the low-qRAM quantum herding attack.

As the second contribution, we also introduce various quantum attacks on the very important hash combiners, including the quantum herding attacks on hash concatenation combiner, Hash-Twice, and Zipper hash functions by exploiting their different features. All our quantum herding attacks not only reduce the qRAM from previous exponential size to our polynomial size, but also reduce the time complexities.

For the quantum preimage attack on hash XOR combiners, we introduce an efficient low-qRAM quantum algorithm to build Leurent and Wang's interchange structure [43]. Then, based on Schrottenloher and Stevens's quantum Meet-in-the-Middle attack [50], we propose a low-qRAM preimage attack on hash XOR combiner by reducing the qRAM $2^{0.143n}$ of previous attack [5] to ours $2^{0.028n}$. Moreover, the time complexity is also reduced from previous $2^{0.495n}$ to ours $2^{0.485n}$.

For hash concatenation combiner, we introduce a low-qRAM quantum collision attack, which significantly reduce the needed qRAM from previous $2^{0.143n}$ to ours $\mathcal{O}(n)$, while the time complexity is also reduced from $2^{0.43n}$ to ours $2^{0.4n}$. All the attacks are summarized in Table 1.

Table 1: A Summary of the Attacks.

Target	Attacks	Settings	Time	qRAM	cRAM	Generic	Ref.
$\mathcal{H}$	Herding	Classical	$2^{0.67n}$	-	$2^{0.33n}$	-	[37]
		Quantum	$2^{0.43n}$	$2^{0.43n}$	-	-	[7]
		Quantum	$2^{0.46n}$	$\mathcal{O}(n)$	$2^{0.23n}$	-	Sect. 4
$\mathcal{H}_1 \oplus \mathcal{H}_2$	Preimage	Classical	$2^{0.83n}$	-	$2^{0.33n}$	2^n	[43]
		Classical	$2^{0.67n}$	-	-	2^n	[21]
		Classical	$2^{0.612n}$	-	$2^{0.61n}$	2^n	[4]
		Quantum	$2^{0.495n}$	$2^{0.143n}$	$2^{0.2n}$	$2^{0.5n}$	[5]
		Quantum	$2^{0.485n}$	$2^{0.028n}$	$2^{0.2n}$	$2^{0.5n}$	Sect. 5
$\mathcal{H}_1 \| \mathcal{H}_2$	Collision	Classical	$2^{0.5n}$	-	-	2^n	[35]
		Quantum	$2^{0.43n}$	$2^{0.143n}$	$2^{0.2n}$	$2^{0.67n}$	[5]
		Quantum	$2^{0.4n}$	$\mathcal{O}(n)$	$2^{0.2n}$	$2^{0.67n}$	Sect. 6
	Herding	Classical	$2^{0.67n}$	-	$2^{0.33n}$	-	[2]
		Quantum	$2^{0.49n}$	$2^{0.143n}$	$2^{0.2n}$	-	[5]
		Quantum	$2^{0.467n}$	$\mathcal{O}(n)$	$2^{0.2n}$	-	Sect. 7
Hash-Twice	Herding	Classical	$2^{0.667n}$	$2^{0.33n}$	-	-	[2]
		Quantum	$2^{0.467n}$	$\mathcal{O}(n)$	$2^{0.2n}$	-	Sect. 8
Zipper	Herding	Classical	$2^{0.667n}$	-	$2^{0.33n}$	-	[2]
		Quantum	$2^{0.467n}$	$\mathcal{O}(n)$	$2^{0.2n}$	-	Sect. 9

2 Preliminaries

2.1 Quantum Computation and Quantum RAM

Superposition Oracles for Classical Circuit. Let the quantum oracle of a function $f : \mathbb{F}_2^m \mapsto \mathbb{F}_2^n$ be the unitary operator $\mathcal{U}_f$ that $\mathcal{U}_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle$

with $x \in \mathbb{F}_2^m$ and $y \in \mathbb{F}_2^n$. When $\mathcal{U}_f$ acts on superposition states, we have

$$\mathcal{U}_f \left(\sum_{x \in \mathbb{F}_2^n} a_i \left| x \right\rangle \left| y \right\rangle \right) = \sum_{x \in \mathbb{F}_2^n} a_i \left| x \right\rangle \left| y \oplus f(x) \right\rangle . \tag{1}$$

Variations on Grover's Algorithm. The task is to find the labeled element from the set X. Suppose we denote the subset of labeled elements by $M \subset X$ and know the fraction of the labeled elements $\epsilon = |M|/|X|$. The classical algorithm to solve this problem needs $O(1/\epsilon)$ iterations. A quantum algorithm can be expressed as a function of two parameters.

- *Setup* operation, i.e., sampling a uniform element from X. Denote the cost (execution time) of *Setup* as $|Setup|_{RT}$.
- *Checking* operation, i.e. checking if an element is labeled. Denote the cost (execution time) of *Checking* as $|Checking|_{RT}$.

Grover's algorithm [30] is a quantum search process for finding the labeled elements, whose complexity is a function of the quantum *Setup* cost $|Setup|_{RT}$ of construction of uniform superposition of all elements from X, and the quantum *Checking* cost $|Checking|_{RT}$. The time complexity of Grover's algorithm is $\sqrt{1/\epsilon} \cdot (|Setup|_{RT} + |Checking|_{RT})$. Assuming the *Setup* and *Checking* steps are simple, Grover's algorithm can find the element $x \in M$ at a cost of $\mathcal{O}(\sqrt{1/\epsilon})$.

Grover's algorithm can also be described as a special case of quantum amplitude amplification (QAA), which is a quantum algorithm introduced by Brassard, Høyer, Mosca, and Tapp [16]. Intuitively, assuming there exists an quantum algorithm $\mathcal{A}$ to produce a superposition of the good subspace and the bad subspace of X. Let a be the initial success probability that the measurement of $\mathcal{A} \left| 0 \right\rangle$ is good. Let $\mathcal{B}$ be a function that classifies the outcomes of $\mathcal{A}$ as either good or bad state. Quantum Amplitude Amplification (QAA) technique achieves the same result as Grover's algorithm with a quadratic improvement. The time complexity of QAA is about

$$\sqrt{1/a} \cdot (|\mathcal{A}|_{RT} + |\mathcal{B}|_{RT}). \tag{2}$$

Quantum Random Access Memories (qRAM). A quantum random access memory (qRAM) is a quantum analogue of a classical random access memory (RAM), which uses n-qubit to address any quantum superposition of 2^n memory cells. Given a list of classical data $L = \{x_0, \cdots, x_{2^n-1}\}$ with $x_i \in \mathbb{F}_2^m$, the qRAM for L is modeled as an unitary transformation $\mathcal{U}_{\mathsf{qRAM}}^L$ such that

$$\mathcal{U}_{\mathsf{qRAM}}^L : \left| i \right\rangle_{\mathsf{Addr}} \otimes \left| y \right\rangle_{\mathsf{Out}} \mapsto \left| i \right\rangle_{\mathsf{Addr}} \otimes \left| y \oplus x_i \right\rangle_{\mathsf{Out}} , \tag{3}$$

where $i \in \mathbb{F}_2^n$, $y \in \mathbb{F}_2^m$, and $\left| \cdot \right\rangle_{\mathsf{Addr}}$ and $\left| \cdot \right\rangle_{\mathsf{Out}}$ may be regarded as the address and output registers respectively. Therefore, we can access any quantum superposition of the data cells by using the corresponding superposition of addresses:

$$\mathcal{U}_{\mathsf{qRAM}}^L \left(\sum_i a_i \left| i \right\rangle \otimes \left| y \right\rangle \right) = \sum_i a_i \left| i \right\rangle \otimes \left| y \oplus x_i \right\rangle . \tag{4}$$

For the time being, it is unknown how a working qRAM (at least for large qRAMs) can be built. Nevertheless, this disappointing fact does not stop researchers from working in a model where large qRAMs are available, in the same spirit that people started to work on classical and quantum algorithms long before a classical or quantum computer had been built. From another perspective, the absence of large qRAMs and the fact that a qRAM of size $O(n)$ can be simulated with a quantum circuit of size $O(n)$ makes it quite meaningful to conduct research in an attempt to reduce or even avoid the use of qRAM in quantum algorithms.

CNS collision finding algorithm [18]. At ASIACRYPT 2017, Chailloux, Naya-Plasencia and Schrottenloher [18] introduced the first quantum collision finding algorithm without exponential size qRAM. Their algorithm is denoted as CNS algorithm in this paper. The time complexity of the algorithm is $\mathcal{O}(2^{2n/5})$, with a quantum memory of $\mathcal{O}(n)$ and a classical memory of $\mathcal{O}(2^{n/5})$. The CNS algorithm is based on a quantum membership algorithm.

Definition 1. *Given a set L of 2^k n-bit strings, a classical membership oracle is a function f_L that computes: $f_L(x) = 1$ if $x \in L$ and 0 otherwise.*

A quantum membership oracle for L is an operator O_L that computes f_L:

$$O_L(|x\rangle |b\rangle) = |x\rangle |b \oplus f_L(x)\rangle .$$

When the set L of size 2^k is stored in some classical memory, Chailloux et al. implement the quantum operator O_L in time $n2^k$ with $2n + 1$ bits of quantum memory. CNS collision finding algorithm can be divided into two parts, i.e., the precomputing part and the matching part.

Precomputing Part: Given a hash function h that $h(m) = T$, the CNS algorithm first builds a table L of size 2^k, where the r-bit most significant bits (MSB) of all $x \in L$ are zero, and store L in a classical memory. The way to build L is to perform 2^k times of Grover's algorithm with time complexity of $2^k \times 2^{r/2} = 2^{k+r/2}$.

The Matching Part: Apply the QAA algorithm. In the setup phase $\mathcal{A}$, the Grover's algorithm is applied to produce a superposition of m, where the r-bit MSBs of m are zero. The time of the setup phase is $|\mathcal{A}|_{RT} = 2^{r/2}$. Then, in the checking phase $\mathcal{B}$, a quantum membership algorithm is applied to classify that if m is in L or not. $|\mathcal{B}|_{RT} = 2^k$. Since the initial probability, that the measurement of $\mathcal{A}|0\rangle$ is good, is $a = \frac{2^k}{2^{n-r}}$ (since only the last $n - r$ bits should be matched). According to Equation (2), time complexity of this part is

$$\sqrt{\frac{2^{n-r}}{2^k}} \cdot (2^{r/2} + 2^k). \tag{5}$$

Totally, the time of the CNS algorithm is

$$\sqrt{\frac{2^{n-r}}{2^k}} \cdot (2^{r/2} + 2^k) + 2^{k+r/2}. \tag{6}$$

By assigning $r = 2k = 2n/5$, Equation (6) is achieve to be optimal, which is $\mathcal{O}(2^{2n/5})$. The quantum memory is used when applying quantum membership algorithm, which is $\mathcal{O}(n)$. The classical memory is $2^{n/5}$ to store L.

In this paper, the CNS algorithm is frequently used. In several applications of our paper, the CNS algorithm is only a local step, so its time complexity in Equation (6) must be traded off against other complexities occur in other steps. To better use the CNS algorithm, we define the matching part as $\mathrm{CNS}_h(m, L)$ for a given table L and h in the following.

Definition 2. *Let $CNS_h(m, L)$ be the matching part of CNS algorithm, which finds m so that $h(m) \in L$. Given the table L of size 2^k stored in classical memory, whose elements are prefixed with r-bit zeros, the time complexity $|CNS_h(m, L)|_{RT} = \sqrt{\frac{2^{n-r}}{2^k}} \cdot (2^{r/2} + 2^k)$.*

Quantum Two-list Merging Algorithm. At CRYPTO 2022, Schrottenloher and Stevens [50] introduced the quantum two-list merging algorithm to build the quantum MitM attack: For a given global guess $G \in \mathbb{F}_2^g$, two small lists are computed and merged to on the fly. Suppose the two small lists are L_1 and L_2, the goal is to determine if there are elements $x \in L_1$ and $y \in L_2$ such that $x = y$ (called a solution). Let $O_{\texttt{merge}}$ be the unitary operator that

$$O_{\texttt{merge}}(|G\rangle |b\rangle) = |G\rangle |b \oplus f(G)\rangle, \text{where } f(G) = \begin{cases} 1 \text{ if a solution occurs} \\ 0 \text{ otherwise} \end{cases}. \quad (7)$$

Lemma 1. *[50] Assume that there exists an implementation of $O_{\texttt{merge}}$ with time complexity T. Then there is a quantum MitM attack with time complexity:*

$$(\frac{\pi}{4} 2^{g/2} + 1) \times T. \quad (8)$$

The T is roughly estimated by

$$\min(|L_1|, |L_2|) + \sqrt{\max(|L_{\texttt{merge}}|, |L_1|, |L_2|)}, \quad (9)$$

where $L_{\texttt{merge}}$ is the merged list. The quantum random access memory needed is of size $\min(|L_1|, |L_2|)$.

2.2 Iterated Hash Constructions

Iterated hash functions $\mathcal{H}(IV, M) = T$ commonly first pad and split the message M into message blocks of fixed length, i.e., $M = m_1 \| m_2 \| \cdots \| m_L$. The message blocks are processed sequentially and iteratively by the compression function h, i.e., $x_i = h(x_{i-1}, m_i)$, where $x_0 = IV$ is a public value, $T = x_L$ is the n-bit digest, the chaining value $x_i \in \mathbb{F}_2^n$. Two commonly used hash functions following the classical Merkle-Damgård construction [19,46] and the HAIFA construction

[8]. In this paper, we only consider the Merkle-Damgård construction and its extensions.

The concatenation combiner $\mathcal{H}_1(IV_1, M)\|\mathcal{H}_2(IV_2, M) = T_1\|T_2$ is one of the most studied hash combiner, that first described by Preneel in 1993 [49]. In 2004, Joux [35] described the multi-collision attack and attack an $2n$-bit output hash combiner with $2^{n/2}$ for collision and 2^n for preimage. Besides the concatenation combiner, there are other constructions:

– The XOR hash combiner $\mathcal{H}_1(IV_1, M) \oplus \mathcal{H}_2(IV_2, M) = T$.
– Hash-Twice is originally defined in [2]: $\mathcal{H}_2(\mathcal{H}_1(IV, M), M) = T$ shown in Figure 1.
– Zipper hash [44] is defined as $\mathcal{H}_2(\mathcal{H}_1(IV, M), \overleftarrow{M}) = T$ shown in Figure 2.

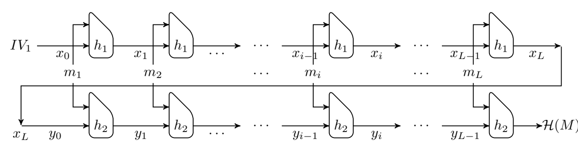


Fig. 1: Hash-Twice Construction

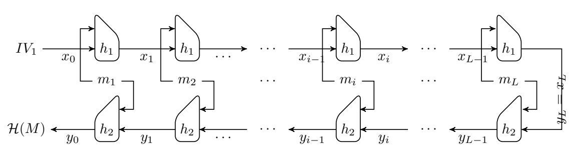


Fig. 2: Zipper Hash Construction

3 Basic techniques and their quantum versions

In this section, we give brief introductions of Joux's multi-collision technique, diamond structure (DS) and their quantum versions.

3.1 Joux's multi-collision

At CRYPTO 2004, Joux [35] introduced an efficient method to build multi-collision on iterated hash functions. As shown in Figure 3, started from x_0, the attacker performs t birthday attacks to find t collisions. Based on the message blocks $m_1, m_2, \cdots m_t$ and $m'_1, m'_2, \cdots m'_t$, the attacker can build 2^t collision message pairs (denoted as 2^t-$\mathcal{M}_{\texttt{MC}}$), e.g., $(m_1 \| m'_2 \| \cdots \| m_t, m'_1 \| m_2 \| \cdots \| m'_t,)$. The time complexity to build the 2^t collision message pairs is $t \cdot 2^{n/2}$. In quantum setting, CNS's algorithm can build one collision in time $2^{2n/5}$. Therefore, the time to build 2^t-$\mathcal{M}_{\texttt{MC}}$ is $t \cdot 2^{2n/5}$. The quantum attack only uses a classical memory $2^{n/5}$.

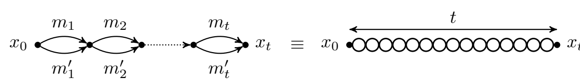


Fig. 3: Joux's multi-collision

3.2 Diamond structure and its New Quantum Algorithm in qRAM-free setting

Kelsey and Kohno in [37] invented the diamond structure. Similar to Joux's multi-collisions and Kelsey and Schneier's expandable message [38], diamond is also a kind of multi-collision. The difference is that, instead of mapping a single starting state to a final state in the form of sequential chain like Joux's multi-collisions, a 2^t-diamond maps a set of 2^t leaf states to a common root state as shown in Figure 4. In classical setting, several improvements [9,39] on building diamond structure have been proposed.

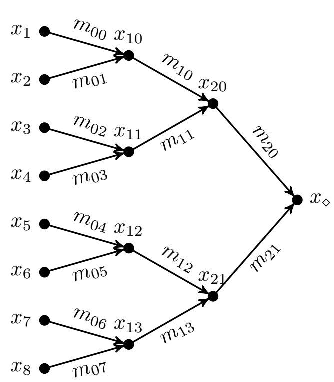


Fig. 4: 2^3-diamond

Bao et al. [5] initially introduced the quantum diamond structure algorithm for both qRAM and qRAM-free scenarios. However, when try to replicate their algorithm, we find their qRAM-free algorithm is incorrect [7].

Later, at ASIACRYPT 2022, Benedikt, Fischlin, Huppert [7] presented a quantum diamond structure algorithm utilizing exponential qRAM, resulting in a time complexity of $t^{1/3} \cdot 2^{(n+2t)/3}$. Consider a level s of the 2^t-diamond structure and try to connect 2^s nodes $\{x_{s,1}, \cdots, x_{s,2^s}\}$ in a pairwise manner. Benedikt et al. split the 2^s nodes into a upper and a lower half of 2^{s-1} nodes each. For the upper half, they compute a list Y of 2^l hash evaluations $h(m_j, x_{s,i})$ with $i = 1, \cdots, 2^{s-1}$, which equally spread out over the 2^{s-1} nodes. Hence, for each node, there are $\frac{2^l}{2^{s-1}}$ hash evaluations. Store Y in qRAM, and apply Grover's algorithm to connect the first value $x_{s,2^{s-1}+1}$ of the lower half to some of these 2^l values with some message block m'. Once a connection message is found, remove the partner node from the upper half and all of its $2^l/2^{s-1}$ entries from Y. Then, add this amount of new values, again equally spread out over the remaining $2^{s-1} - 1$ values paired up, to fill the list Y up to 2^l elements again. Then connect the second note $x_{s,2^{s-1}+2}$ to Y. Continue till all 2^s nodes are connected, then proceed with the next level $s - 1$ until the entire tree is built.

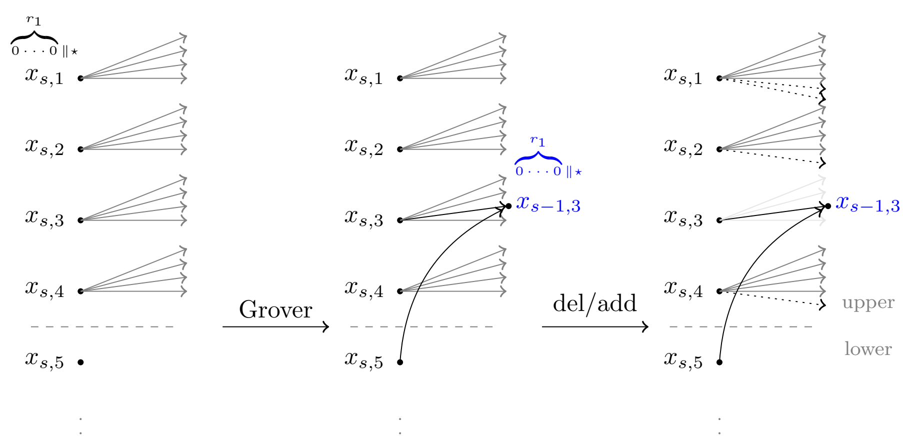


Fig. 5: building diamond

A new qRAM-free quantum algorithm to build the diamond structure. In this section, we adapt Benedikt et al.'s [7] method into a qRAM-free version. As shown in Figure 5, again consider a level s of the 2^t-diamond structure and try to connect 2^s nodes $\{x_{s,1}, \cdots, x_{s,2^s}\}$ in a pairwise manner.

[7] The flaw has been confirmed by the authors of [5] through private communication.

1. Begin with 2^t leaf nodes that share a common suffix of r_0 0s for the purpose of connection.
2. Let's consider a specific level $s \leq t$ of the tree where we aim to connect the 2^s nodes $\{x_{s,1}, ..., x_{s,2^s}\}$ pairwise. Divide the 2^s nodes into two halves, the upper half with 2^{s-1} nodes $\{x_{s,1}, \cdots, x_{s,2^{s-1}}\}$ and the lower half with 2^{s-1} nodes $\{x_{s,2^{s-1}+1}, x_{s,2^{s-1}+2}, \cdots, x_{s,2^s}\}$. For the upper half, compute a list Y of 2^l hash values $h(m_j, x_{s,i})$ with $i = 1, \cdots, 2^{s-1}$, where the r_1 MSBs of $h(m_j, x_{s,i})$ are zero. The 2^l hash values equally spread out over the 2^{s-1} nodes, with $\frac{2^l}{2^{s-1}}$ hash values for each note. Here, similar to CNS algorithm in Section 2.1 to build L whose elements are prefixed with r-bit zero, we also apply Grover's algorithm to build Y. For each note $x_{s,i}$ with $i = 1, \cdots, 2^{s-1}$, run Grover's algorithm to find m_j so that the r_1 MSBs of $h(m_j, x_{s,i})$ are zero. The time to find one m_j is $2^{r_1/2}$. In order to find $\frac{2^l}{2^{s-1}}$ such m_j for node $x_{s,i}$, we apply $\frac{2^l}{2^{s-1}}$ times of Grover's algorithm. Therefore, to build Y, the total time complexity is
$$2^l \times 2^{r_1/2} = 2^{l+\frac{r_1}{2}}. \tag{10}$$
3. Store Y in a classical memory with 2^l elements $(h(m_j, x_{s,i}), m_j, x_{s,i})$ indexed by $h(m_j, x_{s,i})$. For the first node $x_{s,2^{s-1}+1}$ of the lower half, apply CNS algorithm in Section 2.1 to find a message block m' so that $h(m', x_{s,2^{s-1}+1})$ hits one of the entries of Y. According to Definition 2, apply $\mathrm{CNS}_h(m', Y)$ to find such m', whose time complexity is
$$\sqrt{\frac{2^{n-r_1}}{2^l}} \cdot (2^{r_1/2} + 2^l). \tag{11}$$
4. After m' is found, delete the partner node and all of its $2^l/2^{s-1}$ entries from Y. Add $2^l/2^{s-1}$ new values for Y with similar ways to Step 2 to fill Y up to 2^l elements again. Now each note of the upper half corresponds to $2^l/2^{s-1} - 1$ elements. Delete the first node $x_{s,2^{s-1}+1}$ from lower half. The time complexity to fill Y again is
$$2^l/2^{s-1} \times 2^{r_1/2} = 2^{l-s+1+\frac{r_1}{2}}. \tag{12}$$
5. Repeat Step 3 and Step 4 until the lower half is empty. That means all the nodes of the layer of level s have been connected pairwise.

To build the layer of level s, totally, the CNS algorithm in Step 3 is repeated 2^{s-1} times. After the i-th node $x_{s,2^{s-1}+i}$ $(i = 1, \cdots, 2^{s-1} - 1)$ in the lower half has been connected to Y, according to Step 4, $\frac{2^l}{2^{s-1}-(i-1)}$ elements have to be generated to fill Y up to 2^l again, whose time complexity is
$$\frac{2^l}{2^{s-1} - (i-1)} \times 2^{r_1/2}. \tag{13}$$
Therefore, the total time complexity to build the layer of level s is
$$T_s = 2^l \times 2^{r_1/2} + 2^{s-1} \cdot \sqrt{\frac{2^{n-r_1}}{2^l}} \cdot (2^{r_1/2} + 2^l) + \sum_{i=1}^{2^{s-1}-1} \frac{2^l}{2^{s-1} - (i-1)} \times 2^{r_1/2}. \tag{14}$$

To build the 2^t-diamond structure which includes t layers, the total time is

$$\sum_{s=t}^{2} T_s. \tag{15}$$

We could calculate

$$T_s = 2^{s-1}\cdot\sqrt{\frac{2^{n-r_1}}{2^l}}\cdot(2^{r_1/2}+2^l)+2^l\cdot 2^{r_1/2}\cdot\sum_{j=1}^{2^{s-1}}\frac{1}{j} = 2^{s-1}\cdot\sqrt{\frac{2^{n-r_1}}{2^l}}\cdot(2^{r_1/2}+2^l)+O(s\cdot 2^{l+r_1/2})$$

using $\sum_{j=1}^{q}\frac{1}{j} \leq \ln q + c$ for the harmonic series. Then T_s could be minimized to $\mathcal{O}(s^{1/5}\cdot 2^{(2n+4s+4)/5})$ by setting $r_1 = 2l$ and $l = \frac{n+2s+2-2\log_2 s}{5}$.

The final complexity is obtained from summing over all t levels:

$$\begin{aligned}\sum_{s=1}^{t}\mathcal{O}(s^{1/5}\cdot 2^{(2n+4s+4)/5}) \leq & \quad \mathcal{O}(2^{(2n+4+\log_2 t)/5}\cdot\sum_{s=1}^{t}2^{\frac{4s}{5}}) \\ = & \quad \mathcal{O}(2^{(2n+4+\log_2 t)/5}\cdot 2^{\frac{4t}{5}}) \\ = & \quad \mathcal{O}(2^{(2n+4t+4+\log_2 t)/5}),\end{aligned}$$

which is about $\mathcal{O}(2^{(2n+4t)/5})$. The classical memory is dominated by $\mathcal{O}(2^{(n+2t)/5})$ to store Y for the first layer. The size of qRAM is $\mathcal{O}(n)$ when applying CNS algorithm.

4 Herding Attack in Quantum Settings with Low qRAM

The herding attack on iterated hash function is first given by Kelsey and Kohno [37]. In the attack, the adversary chooses a public hash value h_T, and then, she is challenged with a prefix P. Her goal is to find a suffix S such that $h_T = H(P\|S)$. At ASIACRYPT 2022, Benedikt, Fischlin, and Huppert [7] presented the quantum herding attack with $\sqrt[3]{n}\cdot 2^{3n/7}$ on iterated hash function with n-bit digest based on BHT algorithm. Their quantum attack also needs exponentially large quantum memory (qRAM) inherited from the BHT algorithm [17]. Therefore they left an open question on how to devise quantum herding attacks with polynomial size of quantum memory (qRAM). In this section, we answer the open question positively. As shown in Figure 6, our herding attack is consisted of four steps:

- Step 1 is to build a 2^k-diamond structure. In classical herding attack by Kelsey and Kohno [37] and the quantum one by Benedikt et al. [7], the leaves x_i $(1 \leq i \leq 2^k)$ are randomly chosen. In our quantum attack, the r most significant bits (MSB) of x_i are zero.
- Step 2 and Step 3 is to find a single block message M_{link} such that $h(P\|M_{link})$ collides with some value $x_j \in D$.
- Step 4 is to produce the message $M = P\|M_{link}\|M_j$, where M_j is a sequence of message blocks linking x_j to h_T with the diamond structure.

Our quantum herding attack is given in Algorithm 1.

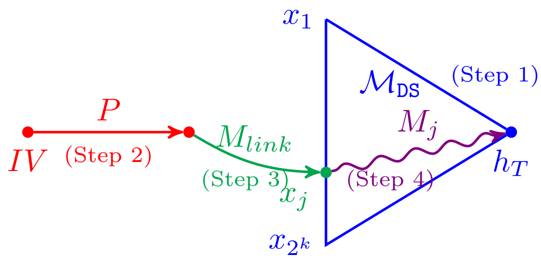


Fig. 6: Herding Attack on Iterated Hash Function

Algorithm 1: Herding Attack on Iterated Hash Function without qRAM

Off-line precomputation: Precompute the diamond structure using CNS quantum collision algorithm. Collect 2^k starting chaining values $D = \{x_1, x_2, \cdots, x_{2^k}\}$, where the r MSBs of $x_i \in \mathbb{F}_2^n$ are zero. The root is denoted as h_T and publish h_T.

On-line precomputation:

begin

Receive the challenged prefix P and compute the chaining value after absorbing the message P: $\bar{x} = \bar{H}(IV, P)$.

`/* Finding the linking message` M_{link} `by applying variant of CNS collision-finding algorithm: */`

Store $D = \{x_1, x_2, \cdots, x_{2^k}\}$ in a classical memory L.

Define $S_r^h := \{(m, h(\bar{x}, m)) : \exists z \in \{0,1\}^{n-r}, h(\bar{x}, m) = \underbrace{0 \cdots 0}_{r\ times} \| z, z \in \{0,1\}^{n-r}\}$, where h is the compression function with n-bit chaining value $\bar{x}$. Let $f_L^h(m) := 1$ if $\exists x' \in L$, $h(\bar{x}, m) = x'$, and $f_L^h(m) := 0$ otherwise.

Apply quantum amplification algorithm:

begin

The setup $\mathcal{A}$ is the construction of $|\phi\rangle := \frac{1}{\sqrt{|S_r^h|}} \sum_{m \in S_r^h} |m, h(\bar{x}, m)\rangle$.

The projector is a quantum oracle query to $O_{f_L^h}$ meaning that

$$O_{f_L^h}(|m, h(\bar{x}, m)\rangle |b\rangle) = |m, h(\bar{x}, m)\rangle |b \oplus O_{f_L^h}(m)\rangle. \tag{16}$$

end

Let $M_{link} = m$ and produce the message: $M = P \| M_{link} \| M_j$, where M_j is a sequence of message blocks linking x_j to h_T following the diamond structure built before.

end

Complexity. The time complexity to build the 2^k diamond structure is $k^{1/5} \cdot 2^{(2n+4k)/5}$ with a classical memory $k^{3/5} \cdot 2^{(n+2k)/5}$ according to Section 3.2. The time complexity of the setup phase is $2^{r/2}$ with Grover algorithm. According to the quantum membership algorithm [18], the time complexity to implement $O_{f_L^h}$ is 2^k. For $(m, h(\bar{x}, m)) \in S_r^h$, $f_L^h(m) = 1$ holds with probability of $2^{k-(n-r)}$. Therefore, about $2^{\frac{n-r-k}{2}}$ calls of A, $A^\dagger$, $O_{f_L^h}$, $O_{f_L^h}^\dagger$ are needed to produce the correct $M_{link} = m$. Hence, the time complexity to find the M_{link} in Line 8 is $2^{\frac{n-r-k}{2}}(2^{r/2} + 2^k)$ with a classical memory 2^k to store L. Hence, the total time complexity is

$$2^{\frac{n-r-k}{2}}(2^{r/2} + 2^k) + k^{1/5} \cdot 2^{(2n+4k)/5}. \tag{17}$$

The classical memory complexity is bounded by the construction of the diamond structure, i.e., $k^{3/5} \cdot 2^{(n+2k)/5}$.

The best-case complexity. The optimal complexity is to balance the three formulas, i.e., $\frac{n-k}{2}$, $\frac{n-r+k}{2}$, and $\frac{2n+4k}{5}$. When $k = n/13$ and $r = 2n/13$, the optimal complexity is achieved which results in $\mathcal{O}(2^{6n/13}) = \mathcal{O}(2^{0.46n})$ time complexity and $\mathcal{O}(2^{3n/13}) = \mathcal{O}(2^{0.23n})$ classical memory.

Remark. Bao et al. [5] also proposed a qRAM-free herding attack based on a flawed method of building the diamond structure as shown in Section 3.2. After correcting with our right algorithm in Section 3.2, Bao et al.'s qRAM-free herding attack needs a time complexity of $\mathcal{O}(2^{14n/29}) = \mathcal{O}(2^{0.48n})$ with a classical memory $\mathcal{O}(2^{7n/29}) = \mathcal{O}(2^{0.24n})$, which is inferior to our attacks.

5 Interchange Structure and Preimage attack on XOR combiners

5.1 Basic Interchange Structure Technique [43]

At EUROCRYPT 2015, Leurent and Wang [43] invented the interchange structure (IS), which is used to devise a preimage attack on the XOR combiner, i.e., $\mathcal{H}_1(IV_1, M) \oplus \mathcal{H}_2(IV_2, M) = T$. The interchange structure contains a set of messages $\mathcal{M}_{\mathtt{IS}}$ and two sets of states $\mathcal{A}$ and $\mathcal{B}$, so that for any state pair $(A_i, B_j | A_i \in \mathcal{A}, B_j \in \mathcal{B})$, the attacker can pick a message $M \in \mathcal{M}_{\mathtt{IS}}$ such that $A_i = \mathcal{H}_1(IV_1, M)$ and $B_j = \mathcal{H}_1(IV_2, M)$. Suppose there is a 2^k-interchange structure (the sizes of $\mathcal{A}$ and $\mathcal{B}$ are both 2^k). In order to reach the target value T, they select a random block m, and evaluate $L_1 = \{A'_i = h_1(A_i, m), i = 1 \cdots 2^k\}$ and $L_2 = \{B'_j = T \oplus h_2(B_j, m), j = 1 \cdots 2^k\}$, where h_1 and h_2 are the compression functions. If there is a match between the two lists L_1 and L_2, then

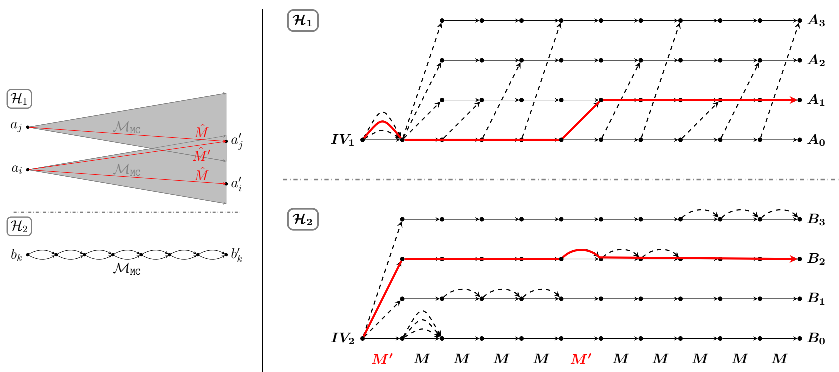
$$h_1(A_i, m) = T \oplus h_2(B_j, m) \Leftrightarrow \mathcal{H}_1(IV_1, M\|m) \oplus \mathcal{H}_2(IV_1, M\|m) = T. \tag{18}$$

The above technique is exact an Meet-in-the-Middle approach. For a given m, it produce the preimage with probability 2^{2k-n} with time complexity 2^k. Therefore, to find the preimage, 2^{n-2k} m should be exhausted with a time complexity of $2^{n-2k} \times 2^t = 2^{n-k}$.

To build a 2^k-interchange structure (the sizes of $\mathcal{A}$ and $\mathcal{B}$ are both 2^k), the classical time complexity is $\tilde{O}(2^{2k+n/2})$ in [43].

5.2 Low qRAM Quantum Version of Interchange Structure

For the hash XOR combiners $\mathcal{H}_1(IV_1, M) \oplus \mathcal{H}_2(IV_2, M) = T$, the basic technique to build interchange structure is to build a single switch. As shown in Figure 7(a), given the multi-collision set $\mathcal{M}_{\texttt{MC}}$ of size 2^t, $\forall M \in \mathcal{M}_{\texttt{MC}}$, $h_2^*(b_k, M) = b'_k$. The single switch algorithm (Alg. 2) is to find a pair $\hat{M}$, $\hat{M}' \in \mathcal{M}_{\texttt{MC}}$, such that $h_1^*(a_j, \hat{M}) = h_1^*(a_i, \hat{M}')$.



(a) Building a switch **(b)** Interchange structure

Fig. 7: Interchange structure and its building block

Complexity of Algorithm 2:

- In Line 1, the time to build 2^t-$\mathcal{M}_{\texttt{MC}}$ is $t \cdot 2^{2n/5}$, with classical memory $2^{n/5}$ by applying CNS algorithm directly.
- In Line 3, with the superposition in Eq. (20), Grover algorithm is applied to determine a $M = (m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t})$, such that the r MSBs of $h_1^*(a_j, M)$ are zero, whose time complexity is $2^{r/2}$. To find 2^x such M, the time complexity is $2^{x+r/2}$. A classical memory of size 2^x is needed to store L_2.
- In Line 6 a), the setup phase is to produce the superposition of $|\phi_r\rangle$, whose time complexity is about $2^{r/2}$.

  In Line 6 b), the projector is a quantum membership checking, whose time complexity is about 2^x. To ensure that there is at least one collision, we have $2^{t-r} \times 2^x \geq 2^{n-r}$, i.e., $t + x \geq n$. The total time complexity is

$$2^{\frac{n-r-x}{2}} \cdot (2^{r/2} + 2^x) + 2^{x+r/2} + t \cdot 2^{2n/5}. \tag{23}$$

  When $x = \frac{r}{2} = \frac{n}{5}$ and $t = \frac{4n}{5}$, we get the optimal time complexity, i.e., $O(\frac{4n}{5} \cdot 2^{2n/5})$. The qRAM to store L_1 is of polynomial size, which is $O(t \cdot n)$. The classical memory used to store L_2 and in Line 1 is $O(2^{n/5})$.

Algorithm 2: Building a Single Switch in Quantum Settings with Low qRAM

---

1 Use the quantum Joux's multi-collision algorithm to build a set $\mathcal{M}_{\texttt{MC}}$ of 2^t messages for h_2^* that link the starting state b_k to the same state b_k', i.e., $\forall M \in \mathcal{M}_{\texttt{MC}}, h_2^*(b_k, M) = b_k'$. The number of message blocks of M is t. Denote the i-th collision message blocks in Joux's multi-collision are (m_i^0, m_i^1), $1 \leq i \leq t$, which are stored in qRAM L_1, whose size is about $O(t \cdot n)$.

2 Given $|l_1, l_2, ..., l_t\rangle$ $1 \leq i \leq t$ and $l_i \in \{0, 1\}$, O_f is the quantum oracle that computes $O_f(|l_1, l_2, ..., l_t\rangle|0\rangle) = |l_1, l_2, ..., l_t\rangle|m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t}\rangle$ by accessing qRAM L_1. Therefore, we can obtain the superposition of Eq. (20)

a) Apply Hadamard H to the first t qubits of $|0\rangle$, we get

$$\sum_{l_1, l_2, ..., l_t \in \{0,1\}} |l_1, l_2, ..., l_t\rangle|0\rangle. \tag{19}$$

b) Apply O_f to the superposition, we get

$$|\phi\rangle = \sum_{l_1, l_2, ..., l_t \in \{0,1\}} |l_1, l_2, ..., l_t\rangle|m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t}\rangle. \tag{20}$$

3 Select 2^x $(x \leq t)$ $M \in \mathcal{M}_{\texttt{MC}}$, where the r MSBs of $a_j' = h_1^*(a_j, M)$ are zero. Store (a_j', M) in classical memory L_2, whose size is about 2^x. Apply Grover algorithm to produce L_2 (combining with Eq. (20)) with complexity of $2^x \cdot 2^{r/2} = 2^{x+r/2}$.

4 Let $M = (m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t}) \in \mathcal{M}_{\texttt{MC}}$, and define $g_{L_2}^{h_1^*}(M) := 1$ if $a_i' = h_1^*(a_i, M) \in L_2$, and $g_{L_2}^{h_1^*}(M) := 0$ otherwise. `// quantum membership checking`

5 Define $S_r^{h_1^*} := \{M : \exists z \in \{0, 1\}^{n-r}, h_1^*(a_i, M) = \underbrace{0 \cdots 0}_{r\ times} \| z, z \in \{0, 1\}^{n-r}, M \in \mathcal{M}_{\texttt{MC}}\}$.

6 Run a variant of CNS algorithm. Apply quantum amplification algorithm (QAA) to determine the collision.

a) The setup phase of QAA is to compute the following superposition together with Eq. (20)

$$|\phi_r\rangle := \frac{1}{\sqrt{|S_r^{h_1^*}|}} \sum_{x \in S_r^{h_1^*}} |M\rangle \tag{21}$$

b) The projector of the QAA is applying quantum oracle $O_{g_{L_2}^{h_1^*}}$, let $M = (m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t})$,

$$O_{g_{L_2}^{h_1^*}} |M\rangle|y\rangle = |M\rangle|y \oplus g_{L_2}^{h_1^*}(M)\rangle \tag{22}$$

5.3 Preimage attack on XOR combiners with Low qRAM

In classical setting, Leurent and Wang [43] built preimage attack on the XOR combiner with an Meet-in-the-Middle approach. Leurent and Wang first built a 2^k-interchange structure (the sizes of $\mathcal{A}$ and $\mathcal{B}$ are both 2^k) as shown in Section 5.1. In this section, in quantum setting, we apply Schrottenloher and Stevens' quantum MitM attack and quantum merging algorithm [50] (also refer to Section 2.1) to perform our quantum attack on XOR combiners. As shown in Section 5.1, the sizes of L_1 and L_2 should be equal in Leurent and Wang's classical attack to achieve the optimal time complexity. However, according to Equation (9), L_1 and L_2 should of different sizes. According to (18), the matching bits are n bits, therefore, the size of $L_{\texttt{merge}}$ that contains messages satisfy (18) is very small when compared to L_1 and L_2. Actually, we only find one preimage, so that $|L_{\texttt{merge}}|$ is about 1. Without loss of generality, we assume $|L_1|$ is bigger. Then (9) is simplified as

$$|L_2| + \sqrt{|L_1|}. \tag{24}$$

To reach an optimal balance, we choose $|L_1| = 2^{2k}$ and $|L_2| = 2^k$, so that the complexity of the quantum merging algorithm is $\mathcal{O}(2^k)$. We denote this kind of interchange structure as $(2^{2k}, 2^k)$-interchange structure, which is built by applying $2^{3k} - 1$ quantum single switches (Algorithm 2) as the following:

1. Build a single switch from (a_0, b_0) to each of (a_0, b_j) $j = 0, ..., 2^k - 1$,
2. For each j, build switches from (a_0, b_j) to all (a_i, b_j) for all $i = 0, ..., 2^{2k} - 1$,
3. To reach the chain (a_i, b_j) from (a_0, b_0), we first find the switch to jump from (a_0, b_0) to (a_0, b_j) in the first step, then find the switch to jump from (a_0, b_j) to (a_i, b_j) in the second step (see Figure 7(b)).

The time complexity is $\mathcal{O}(\frac{4n}{5} \cdot 2^{3k+2n/5})$ with $\mathcal{O}(2^{n/5})$ classical memory to build the $(2^{2k}, 2^k)$-interchange structure.

According to Lemma 1, we first guess the message block $m \in \mathbb{F}_2^g$, and build the two list L_1 and L_2 with $|L_1| = 2^{2k}$ and $|L_2| = 2^k$, then build the $O_{\texttt{merge}}$ with complexity $\mathcal{O}(2^k)$ according to Equation (24). To find at least one preimage, we have $2^{g+k+2k} = 2^n$, so that $g = n - 3k$. According to Equation (8), the time complexity of the quantum MitM attack is about $2^{\frac{n-3k}{2}} \times 2^k = 2^{\frac{n-k}{2}}$. The qRAM needed in the quantum MitM attack is $|L_2| = 2^k$.

The overall time complexity including the time to build $(2^{2k}, 2^k)$-interchange structure and the quantum MitM attack is $\frac{4n}{5} \cdot 2^{3k+2n/5} + 2^{\frac{n-k}{2}}$. The optimal complexity is $2^{17n/35} = 2^{0.485n}$ by setting $k = n/35$. The classical memory is $\mathcal{O}(2^{n/5})$. The qRAM is $2^{n/35} = 2^{0.0285n}$.

6 Collision attack on Concatenation Combiners in Quantum Settings

For a hash concatenation combiner $\mathcal{H}_1(IV_1, M) \| \mathcal{H}_2(IV_2, M) = T_1 \| T_2$, the collision attack is to find two distinct M and M', so that $\mathcal{H}_1(IV_1, M) \| \mathcal{H}_2(IV_2, M) =$

$\mathcal{H}_1(IV_1, M')\|\mathcal{H}_2(IV_2, M')$. Classically, based Joux's multi-collision method [35], the collision attack can be built in $O(2^{n/2})$. Here, we introduce a new quantum collision attack on the hash combiners in Algorithm 3.

Complexity of Alg. 3. Alg. 3 is quite similar to Alg. 2. When we let $t = n$, $x = 2^{n/5}$, $r = 2^{2n/5}$, the attack is optimal. The time complexity is $n \cdot 2^{2n/5}$ with a classical memory of $2^{n/5}$ and polynomial size of qRAM (n^2).

7 Herding Attack on Concatenation Combiners in Quantum Setting

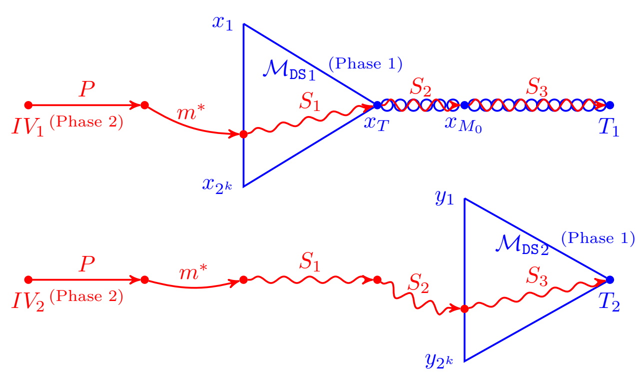


Fig. 8: Herding Attack on Concatenation Combiners in quantum settings

The herding attack on concatenation combiners in quantum settings is given in Figure 8 and Algorithm 4.

Complexity of Algorithm 4.

– In the off-line precompuation phase (Line 2 to 6), the time complexity to build $\mathcal{M}_{\mathtt{DS1}}$, $\mathcal{M}_{\mathtt{MC}_s}$, $\mathcal{M}_{\mathtt{MC}_\ell}$, and $\mathcal{M}_{\mathtt{DS2}}$ is

$$2^{k+2n/5} + t \cdot 2^{2n/5} + 4nk/5 \cdot 2^{2n/5} + 2^{k+2n/5} \approx 2^{k+2n/5},$$

where $t = O(n)$.

– In the online phase (Line 8 to 18), the time to find m^* and S_2 are both $2^{\frac{n-r-k}{2}}(2^{r/2} + 2^k)$.

Therefore, the overall optimal time complexity of Algorithm 4 is $O(2^{7n/15})$ by balancing the off-line and on-line computation phases and assigning $k = n/15$, $r = 2k$, and $t = n$. The memory cost is dominated by building Joux's multi-collision with CNS, i.e., $O(2^{n/5})$ classical memory and $O(n)$ qRAM.

Algorithm 3: Collision attack on Concatenation combiners in Quantum Settings with Low qRAM

1 Use the quantum Joux's multi-collision algorithm to build a set $\mathcal{M}_{\texttt{MC}}$ of 2^t messages for $\mathcal{H}_2$ that link the starting state IV_2 to the same state T_2, i.e., $\forall M \in \mathcal{M}_{\texttt{MC}}, \mathcal{H}_2(IV_2, M) = T_2$. The block length of M is t. Denote the i-th collision message blocks in Joux's multi-collision are (m_i^0, m_i^1), $1 \le i \le t$. Store (m_i^0, m_i^1) in qRAM L_1 (to be used in the construction of superposition), whose size is about $O(t \cdot n)$.

2 Given $|l_1, l_2, ..., l_t\rangle$ $1 \le i \le t$ and $l_i \in \{0,1\}$, O_f is the quantum oracle that computes $O_f(|l_1, l_2, ..., l_t\rangle|0\rangle) = |l_1, l_2, ..., l_t\rangle|m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t}\rangle$ by accessing qRAM L_1. Therefore, we can obtain the superposition of Eq. (26)

a) Apply Hadamard H to the first t qubits of $|0\rangle$, we get

$$\sum_{l_1,l_2,...,l_t \in \{0,1\}} |l_1, l_2, ..., l_t\rangle|0\rangle. \tag{25}$$

b) Apply O_f to the superposition, we get

$$|\phi\rangle = \sum_{l_1,l_2,...,l_t \in \{0,1\}} |l_1, l_2, ..., l_t\rangle|m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t}\rangle. \tag{26}$$

3 Select 2^x $(x \le t)$ $M \in \mathcal{M}_{\texttt{MC}}$, where the r MSBs of $T_1 = \mathcal{H}_1(IV_1, M)$ are zero. Store (T_1, M) in classical memory L_2, whose size is about 2^x. L_2 is produced by applying Grover algorithm and combining with Eq. (26). The time complexity is $2^x \cdot 2^{r/2} = 2^{x+r/2}$.

4 Let $M = (m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t}) \in \mathcal{M}_{\texttt{MC}}$, and define $g_{L_2}^{\mathcal{H}_1}(M) := 1$ if $y = \mathcal{H}_1(IV_1, M) \in L_2$, and $g_{L_2}^{\mathcal{H}_1}(M) := 0$ otherwise. `/* The quantum membership algorithm. */`

5 Define $S_r^{\mathcal{H}_1} := \{M : \exists z \in \{0,1\}^{n-r}, \mathcal{H}_1(IV_1, M) = \underbrace{0 \cdots 0}_{r\ times} \| z, z \in \{0,1\}^{n-r}, M \in \mathcal{M}_{\texttt{MC}}\}$.

6 `/* Run a variant of CNS algorithm. Apply quantum amplification algorithm (QAA). */`

7 The setup phase of QAA is the construction

$$|\phi_r\rangle := \frac{1}{\sqrt{|S_r^{\mathcal{H}_1}|}} \sum_{x \in S_r^{\mathcal{H}_1}} |M\rangle \tag{27}$$

8 The projector of the QAA is applying quantum oracle $O_{g_{L_2}^{\mathcal{H}_1}}$, let $M = (m_1^{l_1}, m_2^{l_2}, ..., m_t^{l_t})$,

$$O_{g_{L_2}^{\mathcal{H}_1}}|M\rangle|y\rangle = |M\rangle|y \oplus g_{L_2}^{\mathcal{H}_1}(M)\rangle \tag{28}$$

Algorithm 4: Quantum Herding Attack on Concatenation Combiners with low qRAM

```
Off-line precomputation:
begin
    Build a diamond $\mathcal{M}_{\texttt{DS1}}$ for $\mathcal{H}_1$, which starts from $2^k$ states $D_1 = \{x_i\}_1^{2^k}$,
      where the $r$ MSBs of $x_i \in \mathbb{F}_2^n$ are zero. To build $\mathcal{M}_{\texttt{DS1}}$, we do not use the
      method given in Section 3.2, but only use CNS algorithm to build each
      collision until the root $x_T$ is derived. Totally,
      $2^{k-1} + 2^{k-2} + \cdots + 1 = 2^k - 1$ times of CNS are applied with time
      complexity $2^{k+2n/5}$ and memory complexity of $2^{n/5}$. The root is $x_T$.
      From the hash value $x_T$, build a $2^t$-Joux's multi-collision $\mathcal{M}_{\texttt{MC}_s}$, in which
      all messages map $x_T$ to a state $x_{M_0}$. Continue to build a $2^{k\cdot\frac{4n}{5}}$-Joux's
      multi-collision $\mathcal{M}_{\texttt{MC}_\ell}$ (consists of $k$ fragments and each fragment is of
      length $4n/5$) on $\mathcal{H}_1$ from the starting state $x_{M_0}$ and mapping to the state
      $T_1$. Denote the terminal states of each of the $k$ fragments of $\mathcal{M}_{\texttt{MC}_\ell}$ by $x_{M_i}$
      for $i$ from 1 to $k$ (note that $x_{M_k} = T_1$).
    Build a diamond $\mathcal{M}_{\texttt{DS2}}$ for $\mathcal{H}_2$, which starts from $2^k$ states $D_2 = \{y_i\}_1^{2^k}$,
      where the $r$ MSBs of $y_i \in \mathbb{F}_2^n$ are zero.. The messages used to building
      $\mathcal{M}_{\texttt{DS2}}$ are all chosen from the set $\mathcal{M}_{\texttt{MC}_\ell}$. For example, the messages
      mapping the first layer of $2^k$ states to the $2^{k-1}$ states in $\mathcal{M}_{\texttt{DS2}}$ are chosen
      from the set of $2^{4n/5}$ messages in the first fragment of $\mathcal{M}_{\texttt{MC}_\ell}$ mapping
      $x_{M_0}$ to $x_{M_1}$. To build $\mathcal{M}_{\texttt{DS2}}$, we do not use the method given in Section
      3.2, but only apply $2^k - 1$ times CNS algorithm variant given by
      Algorithm 2 to find $2^k - 1$ collisions in $\mathcal{M}_{\texttt{MC}_\ell}$. Note that Algorithm 2 is
      exactly the method to find two messages from a set of multi-collisions
      that make two states collides (as shown in Figure 7(a)). The time to
      build $\mathcal{M}_{\texttt{DS2}}$ is $O(2^{k+2n/5})$ with a classical memory $2^{n/5}$.
    Commit $T_1 \| T_2$ to the public.
end
On-line phase:
begin
    Receive the challenged prefix $P$ and compute the internal chaining value
      $x_P = h_1^*(IV_1, P)$ and $y_P = h_2^*(IV_2, P)$.
    /* Finding the linking message $m^*$ by applying variant of CNS
       collision-finding algorithm:                                    */
    Store $D_1$ in a classical memory $L_1$.
    Apply Line 6 to 12 of Algorithm 1 to determine linking message $m^*$ that
      maps $x_P$ to one of the leaf state $x_j$ of $\mathcal{M}_{\texttt{DS1}}$, and retrieve the message $S_1$
      that link the leaf $x_j$ to the root $x_T$.
    Compute $y_T = h_2^*(IV_2, P \| m^* \| S_1)$.
    /* Finding the linking message $S_2$ by applying variant of CNS
       collision-finding algorithm:                                    */
    Store $D_2$ in a classical memory $L_2$.
    Apply CNS algorithm variant given by Algorithm 2 to find $S_2 \in \mathcal{M}_{\texttt{MC}_s}$,
      which maps $y_T$ to one of the leaf state $y_j$ of $\mathcal{M}_{\texttt{DS2}}$, and retrieve the
      message $S_3$ that link the leaf $y_j$ to the root $T_2$.
    $M = P \| m^* \| S_1 \| S_2 \| S_3$ is the returned message.
end
```

8 Quantum Herding attack on Hash-Twice

The attack on Hash-Twice shares the fundamental ideas of the attack on the concatenation combiners, as depicted in Figure 9. The attacker selects T_2 as their commitment and subsequently faces a challenge involving an unknown prefix P. The attack is the same to the attack on concatenation combiner. Please see Algorithm 4 for details. The only difference is that the IV_2 is replaced by T_1. Therefore, the overall optimal time complexity is also $O(2^{7n/15})$ with a classical memory of $O(2^{n/5})$ and a qRAM of $O(n)$.

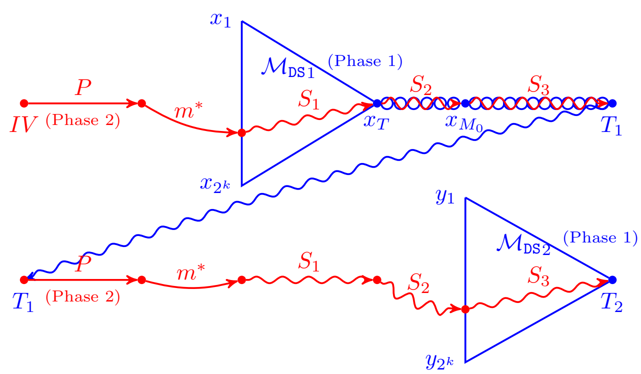


Fig. 9: Herding attack on Hash-Twice

9 Quantum Herding Attack on Zipper Hash

As stated by Andreeva et al. [2], the traditional herding attack with a prefix P can not be applied to Zipper Hash. Therefore, Andreeva et al. [2] gave a variant of the herding attack, where the challenge is placed at the end: as shown in Figure 10, the adversary commits to a hash value h_T, then she is challenged with a suffix S, and has to produce $S_1 \| m^*$ such that $\mathcal{H}(IV, S_1 \| m^* \| S) = h_T$. The complexity of Andreeva et al.'s classical attack is $O(2^{2n/3})$.

In this section, we introduce a quantum version Andreeva et al.'s attack in Algorithm 5. The complexity of the off-line phase dominated by building $\mathcal{M}_{\mathtt{DS}}$, which is about $O(2^{k+2n/5})$. The on-line phase is $2^{\frac{n-r-k}{2}} \cdot (2^{r/2} + 2^k)$ with $t = n$. Let $k = \frac{n}{15}$, $r = 2k$, the optimal complexity is achieved to be $2^{7n/15}$. The memory is $2^{n/5}$.

Conclusion

This paper evaluated the quantum attacks on iterated hash functions and various important hash combiners. Most of the attacks only require polynomial

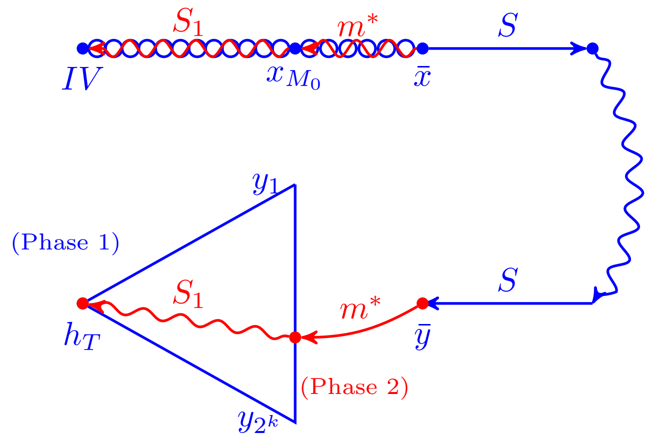


Fig. 10: Herding attack on Zipper Hash

sizes of quantum random access memory (qRAM), or significantly reduce the qRAM from previous $2^{0.143n}$ to $2^{0.028n}$. Since the existence of large qRAM is still questionable, buiding quantum attacks with low-qRAM is of practical relevance. Since for hash functions, the attackers do not need online superposition queries, quantum attacks on hash functions are more friendly than on other keyed primitives like block ciphers. Therefore, exploring the quantum attacks on hash functions is of more practical relevance.

References

1. Elena Andreeva, Charles Bouillaguet, Orr Dunkelman, Pierre-Alain Fouque, Jonathan J. Hoch, John Kelsey, Adi Shamir, and Sébastien Zimmer. New second-preimage attacks on hash functions. *J. Cryptol.*, 29(4):657–696, 2016.
2. Elena Andreeva, Charles Bouillaguet, Orr Dunkelman, and John Kelsey. Herding, second preimage and trojan message attacks beyond merkle-damgård. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, volume 5867 of *Lecture Notes in Computer Science*, pages 393–414. Springer, 2009.
3. Elena Andreeva, Charles Bouillaguet, Pierre-Alain Fouque, Jonathan J. Hoch, John Kelsey, Adi Shamir, and Sébastien Zimmer. Second preimage attacks on dithered hash functions. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 270–288. Springer, 2008.
4. Zhenzhen Bao, Itai Dinur, Jian Guo, Gaëtan Leurent, and Lei Wang. Generic attacks on hash combiners. *J. Cryptol.*, 33(3):742–823, 2020.
5. Zhenzhen Bao, Jian Guo, Shun Li, and Phuong Pham. Evaluating the security of merkle-damgård hash functions and combiners in quantum settings. In Xingliang

Algorithm 5: Quantum Herding attack on Zipper Hash with Low qRAM

1 **Off-line phase: begin**
2 Build a $2^{k \cdot \frac{4n}{5}}$-Joux's multi-collision $\mathcal{M}_{\texttt{MC1}}$ (consists of k fragments and each fragment is of length $4n/5$) that link IV and x_{M_0}. Denote the terminal states of each of the k fragments of $\mathcal{M}_{\texttt{MC1}}$ by x_{M_i} for i from $k-1$ to 0.
3 Build 2^t-Joux's multi-collision $\mathcal{M}_{\texttt{MC2}}$ from x_{M_0} to h_1.
4 Build $\mathcal{M}_{\texttt{DS}}$, which starts from 2^k leaf states $D = \{y_i\}_1^{2^k}$ to the root state h_T, where the r MSBs of $y_i \in \mathbb{F}_2^n$ are zero. Similar to Line 4, we apply $2^k - 1$ times of Algorithm 2 to build $\mathcal{M}_{\texttt{DS}}$, which needs $2^{k+2n/5}$ time and $2^{n/5}$ memory.
5 Commit h_T.
6 **end**
7 **On-line phase: begin**
8 Given the suffix S, compute $\bar{y} = h_2^*(h_1^*(\bar{x}, S), \overleftarrow{S})$.
9 Apply the variant of CNS to find the $m^* \in \mathcal{M}_{\texttt{MC2}}$ to connect $\bar{y}$ with the y_j one of the leaf states of $\mathcal{M}_{\texttt{DS}}$, and retrieve the corresponding message $S_1 \in \mathcal{M}_{\texttt{MC2}}$.
10 Output the message $S_1 \| m^* \| S$.
11 **end**

Yuan, Guangdong Bai, Cristina Alcaraz, and Suryadipta Majumdar, editors, *Network and System Security - 16th International Conference, NSS 2022, Denarau Island, Fiji, December 9-12, 2022, Proceedings*, volume 13787 of *Lecture Notes in Computer Science*, pages 687–711. Springer, 2022.

6. Zhenzhen Bao, Lei Wang, Jian Guo, and Dawu Gu. Functional graph revisited: Updates on (second) preimage attacks on hash combiners. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 404–427. Springer, 2017.
7. Barbara Jiabao Benedikt, Marc Fischlin, and Moritz Huppert. Nostradamus goes quantum. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part III*, volume 13793 of *Lecture Notes in Computer Science*, pages 583–613. Springer, 2022.
8. Eli Biham and Orr Dunkelman. A framework for iterative hash functions - HAIFA. *IACR Cryptol. ePrint Arch.*, page 278, 2007.
9. Simon R. Blackburn, Douglas R. Stinson, and Jalaj Upadhyay. On the complexity of the herding attack and some related attacks on hash functions. *Des. Codes Cryptogr.*, 64(1-2):171–193, 2012.
10. Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Infor-*

mation Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I, pages 552–583, 2019.
11. Xavier Bonnetain and Samuel Jaques. Quantum period finding against symmetric primitives in practice. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):1–27, 2022.
12. Xavier Bonnetain, Gaëtan Leurent, María Naya-Plasencia, and André Schrottenloher. Quantum linearization attacks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 422–452. Springer, 2021.
13. Xavier Bonnetain and María Naya-Plasencia. Hidden shift quantum cryptanalysis and implications. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 560–592. Springer, 2018.
14. Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. On quantum slide attacks. In *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, pages 492–519, 2019.
15. Xavier Bonnetain, André Schrottenloher, and Ferdinand Sibleyras. Beyond quadratic speedups in quantum attacks on symmetric schemes. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 315–344. Springer, 2022.
16. Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
17. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *LATIN '98: Theoretical Informatics, Third Latin American Symposium, Campinas, Brazil, April, 20-24, 1998, Proceedings*, pages 163–169, 1998.
18. André Chailloux, María Naya-Plasencia, and André Schrottenloher. An efficient quantum collision search algorithm and implications on symmetric cryptography. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, pages 211–240, 2017.
19. Ivan Damgård. A design principle for hash functions. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 416–427, 1989.
20. Tim Dierks and Christopher Allen. The tls protocol version 1.0. Technical report, 1999.
21. Itai Dinur. New attacks on the concatenation and XOR hash combiners. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 484–508. Springer, 2016.

22. Xiaoyang Dong, Bingyou Dong, and Xiaoyun Wang. Quantum attacks on some feistel block ciphers. *Des. Codes Cryptogr.*, 88(6):1179–1203, 2020.
23. Xiaoyang Dong, Jian Guo, Shun Li, and Phuong Pham. Triangulating rebound attack on aes-like hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 94–124. Springer, 2022.
24. Xiaoyang Dong, Zhiyu Zhang, Siwei Sun, Congming Wei, Xiaoyun Wang, and Lei Hu. Automatic classical and quantum rebound attacks on aes-like hashing by exploiting related-key differentials. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 241–271. Springer, 2021.
25. Antonio Flórez-Gutiérrez, Gaëtan Leurent, María Naya-Plasencia, Léo Perrin, André Schrottenloher, and Ferdinand Sibleyras. New results on gimli: Full-permutation distinguishers and improved collisions. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 33–63. Springer, 2020.
26. Alan Freier, Philip Karlton, and Paul Kocher. The secure sockets layer (ssl) protocol version 3.0. Technical report, 2011.
27. Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Physical Review A*, 78(5):052310, 2008.
28. Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.
29. Lorenzo Grassi, María Naya-Plasencia, and André Schrottenloher. Quantum algorithms for the k -xor problem. In *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, pages 527–559, 2018.
30. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219, 1996.
31. Akinori Hosoyamada and Yu Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, pages 198–218, 2018.
32. Akinori Hosoyamada and Yu Sasaki. Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. *IACR Cryptology ePrint Archive*, 2020:213, 2020.
33. Akinori Hosoyamada and Yu Sasaki. Quantum collision attacks on reduced SHA-256 and SHA-512. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 616–646. Springer, 2021.
34. Gembu Ito, Akinori Hosoyamada, Ryutaroh Matsumoto, Yu Sasaki, and Tetsu Iwata. Quantum chosen-ciphertext attacks against feistel ciphers. In Mitsuru

Matsui, editor, *Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings*, volume 11405 of *Lecture Notes in Computer Science*, pages 391–411. Springer, 2019.

35. Antoine Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 306–316. Springer, 2004.
36. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 207–237, 2016.
37. John Kelsey and Tadayoshi Kohno. Herding hash functions and the nostradamus attack. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 183–200. Springer, 2006.
38. John Kelsey and Bruce Schneier. Second preimages on n-bit hash functions for much less than 2 n work. In *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*, pages 474–490. Springer, 2005.
39. Tuomas Kortelainen and Juha Kortelainen. On diamond structures and trojan message attacks. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 524–539. Springer, 2013.
40. Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*, pages 2682–2685, 2010.
41. Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*, pages 312–316, 2012.
42. Gregor Leander and Alexander May. Grover meets simon - quantumly attacking the FX-construction. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, pages 161–178, 2017.
43. Gaëtan Leurent and Lei Wang. The sum can be weaker than each part. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 345–367. Springer, 2015.
44. Moses D. Liskov. Constructing an ideal hash function from weak ideal compression functions. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptog-*

raphy, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers, volume 4356 of *Lecture Notes in Computer Science*, pages 358–375. Springer, 2006.

45. Florian Mendel, Christian Rechberger, and Martin Schläffer. MD5 is weaker than weak: Attacks on concatenated combiners. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2009.
46. Ralph C. Merkle. A certified digital signature. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 218–238, 1989.
47. María Naya-Plasencia and André Schrottenloher. Optimal merging in quantum k-xor and k-sum algorithms. *IACR Cryptology ePrint Archive*, 2019:501, 2019.
48. NIST. The post quantum project. https://csrc.nist.gov/projects/post-quantum-cryptography.
49. Bart Preneel. *Analysis and design of cryptographic hash functions*. PhD thesis, Katholieke Universiteit te Leuven Leuven, 1993.
50. André Schrottenloher and Marc Stevens. Simplified MITM modeling for permutations: New (quantum) attacks. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*, volume 13509 of *Lecture Notes in Computer Science*, pages 717–747. Springer, 2022.
51. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134, 1994.
52. Daniel R. Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.
53. Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 570–596. Springer, 2017.
54. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.
55. Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.