Improved Quantum Attacks on Iterated Even-Mansour Ciphers with Classical Queries

Mathieu Degré, Alisée Lafontaine, Aurel Pichollet--Mugnier, and André Schrottenloher

Univ Rennes, Inria, CNRS, IRISA, Rennes, France
firstname.lastname@inria.fr

Abstract. The Even-Mansour cipher is a construction of a keyed pseudorandom permutation from a random unkeyed permutation. Its generalization to multiple rounds, known as *iterated Even-Mansour* or *key-alternating cipher*, is an important abstraction in block cipher design. Yet, while the security of single-round Even-Mansour is tight in the quantum setting (with attacks matching security proofs), much less is known on multi-round versions.

In this paper we study the quantum security of iterated Even-Mansour ciphers with two keys (the key-schedule alternates between two independent keys), which model concrete block ciphers like LED. We give the first quantum attacks improving asymptotically over exhaustive key search for 4 to 6 rounds.

On 4 rounds, we present three attacks: two using collision search and one using a quantum version of the *multibridge* attack of Dinur, Dunkelman, Keller and Shamir (ASIACRYPT 2014), which relies on a quantum walk. The latter reaches up to a quantum time $2^{7n/9}$ where n is the block size, compared to 2^n of exhaustive key search.

On 6 rounds, we present an attack of quantum time $2^n/\sqrt{\log n}$, adapting classical attacks based on multicollisions. In both cases, our new attacks require only classical known-plaintext queries.

Keywords: Even-Mansour cipher · Quantum cryptanalysis · Multibridge attack · Quantum walk

1 Introduction

Over the last years, the advances in quantum computing have put an increasing threat over cryptographic primitives, whose security were mainly studied in the classical setting. This has been particularly impactful for public-key cryptography, where Shor's algorithm [37] breaks most "pre-quantum" schemes standardized before the 2010s.

In symmetric cryptography, Grover's quantum search algorithm [21] seems to be the main threat, but it is limited to a generic quadratic speedup over classical exhaustive key search. This warrants some confidence in the post-quantum security of block ciphers, sometimes under key-size extensions. However, when

lacking explicit quantum security proofs, it is necessary to study the potential specialized attacks in order to keep confidence in the security of symmetric primitives. As the future of quantum computing is uncertain, this task involves studying the various dedicated attacks in all the possible quantum adversarial models.

In order to better understand the behavior of block-ciphers, we are drawn to study simplified models of actual constructions. The Even-Mansour cipher [19] is such an abstraction of real-world cryptographic primitives. Studying its security yields fundamental and practical insights on the security of a wide category of block ciphers. Given a public n-bit permutation Π and n-bit keys k_0 and k_1, the Even-Mansour cipher is defined as in Figure 1:

$$E(x) := k_1 \oplus \Pi(x \oplus k_0) \ .$$

We denote this particular construction as 2-key 1-EM, as opposed to 1-key 1-EM where $k_1 = k_0$.

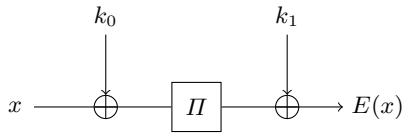

Fig. 1. Even-Mansour cipher (2-key 1-EM).

The 2-key 1-EM cipher is proven secure in the classical setting when Π is a permutation drawn uniformly at random; the so-called *random permutation model*, which is also used in the broader context of iterated constructions that we consider in this paper. Under a data limitation D, attacks reach a time complexity given by $TD = N$, where $N := 2^n$, both in the chosen-plaintext and known-plaintext settings [15,18]. Furthermore, the simplified 1-key 1-EM (where $k_0 = k_1$) achieves the same bound.

Quantum Security of 1-EM. In the quantum setting, the security of this construction has also been well studied. In fact, it was subject to one of the very first dedicated quantum cryptanalytic attacks, by Kuwakado and Morii [30], showing that in the *superposition query* model, it can be broken in polynomial time. This model, known as *Q2* in the literature, assumes that a quantum adversary can query a quantum oracle for the black-box cipher with a secret key. Simon's algorithm [38] can then be used to recover the key. In the Q1 setting, where the adversary only makes classical queries, it was later shown to be secure [1] up to $T^2D = 2^n$, matching the best known attacks. A summary of known attacks is given in Table 1.

One can already notice (this will be important later) that there is currently no optimal attack on 1-round Even-Mansour which uses both classical memory and known-plaintext queries. While the offline-Simon algorithm [8] reaches

the optimal time-data trade-off using only classical memory (in addition to a polynomial number of qubits), it uses chosen-plaintext queries. Conversely the collision-based attack of Kuwakado and Morii [30] can use only known-plaintext queries, but requires quantum-accessible memory (QRACM). The only attack that can do both is from [23], but reaches a suboptimal trade-off.

Table 1. Quantum attacks on 1-key 1-EM and 2-key 1-EM ciphers. Complexities are asymptotic and polynomial factors are omitted.

Adversary	Time-data trade-off	Optimal	Memory	Reference
Q2		$T = \mathsf{polylog}(N)$	Polynomial	Simon [30]
Q1 (CP)	$D + N^{1/2}D^{-1/2}$	$\begin{cases} T = N^{1/3} \\ D = N^{1/3} \end{cases}$	D classical	Offline-Simon [8]
Q1 (KP)	$D + N^{1/2}D^{-1/2}$	$\begin{cases} T = N^{1/3} \\ D = N^{1/3} \end{cases}$	D QRACM	Collision search [30]
Q1 (KP)	$N^{1/2}$	$\begin{cases} T = N^{1/2} \\ D = 2 \end{cases}$	Polynomial	Grover's search
Q1 (KP)	$D + N^{1/2}D^{-1/6}$	$\begin{cases} T = N^{3/7} \\ D = N^{3/7} \end{cases}$	D classical	[23]

Quantum Security of Key-Alternating Ciphers. A natural generalization of 1-EM, which becomes closer to practical block ciphers like the AES, is the *iterated Even-Mansour* cipher, also called the *Key-alternating Cipher* (KAC).

The $r+1$-key r-EM cipher is built from r random permutations $\Pi_1, \ldots, \Pi_r$ and $r+1$ keys selected independently and uniformly at random, and defined as:

$$E(x) = k_r \oplus \Pi_r(k_{r-1} \oplus \ldots \Pi_1(k_0 \oplus x)) \ .$$

If all the keys are independent, this construction is classically indistinguishable from a random permutation until $2^{n\frac{r}{r+1}}$ queries [14], and a matching attack exists [6]. Some very recent works have also aimed at closing the gap between attacks and proofs of security in the quantum setting [5]. However, these attacks are not efficient in time. Besides, constructions based on iterated EM ciphers rarely use independent keys for every round, nor different permutations. Therefore, simplified versions of this generic design have also been studied.

Quantum Security of 2-key 4-EM. In this work, we consider in particular the 2-key 4-round iterated Even-Mansour cipher, denoted 2-key 4-EM, and displayed in Figure 2.

As an example, this abstract construction can be used to model the 128-bit key version of the LED block cipher [22], which uses this exact key schedule

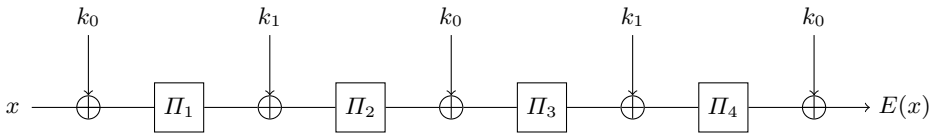

Fig. 2. Two-key iterated Even-Mansour cipher with 4 rounds.

(the master key is divided into two halves of 64 bits) and uses 4-round AES-like permutations with different round constants at each step. Such a key schedule is also used in low-latency ciphers like Qarmav2 [4] or Princev2 [10].

To analyze key-recovery attacks on 2-key 4-EM, one can first notice that by guessing the key k_0, the construction becomes a 1-key 1-EM construction with new permutations. Classical attacks with known-plaintext queries can therefore be transferred to the 2-key 4-EM setting, and in particular, one can reach easily a time-data trade-off $TD = N^2$ until $D = \sqrt{N}$.

In order to decrease the time complexity further, Dinur, Dunkelman, Keller and Shamir introduced the multibridge attack [17]. The multibridge allows one to make use of more data, achieving the trade-off $TD = N^2$ until $D = N$.

Being a natural target for extensions of quantum attacks on 1-EM, the 2-key 4-EM has been studied in the quantum setting by Anand, Ghosh, Isobe and Shiba [3]. They studied attacks based on the offline-Simon algorithm [8] as well as a quantum version of the multibridge attack. However, the attacks proposed in [3], as reported in Table 2, offer only single time-data trade-off points. In particular, there was no attack asymptotically improving over the bound $T = N$ in the Q1 setting.

Quantum Security of Variants with More Rounds. After 4-round variants, the next natural target for key-recovery would be a 6-round variant with two keys (i.e., 2-key 6-EM, in our notation). Such variants have also been studied extensively in the classical setting. One can start by studying an iterated Even-Mansour cipher with 2 rounds and the same key, i.e., 1-key 2-EM, which can then be extended to 2-key 6-EM similarly to the 4-round case.

There are many results in the classical setting, like [16,25,32], which mainly use two techniques: *multicollisions* [16] and *3-XOR algorithms* [32]. Their common point is that they improve on exhaustive search only by a polynomial factor.

In the quantum setting, none of these techniques has been adapted so far. Offline-Simon and Grover-meet-Simon [31] being unapplicable, no attack improving strictly on exhaustive search is known, whether in the Q1 or Q2 setting.

Contributions. In this paper, we give new quantum attacks on iterated Even-Mansour ciphers with two keys in the Q1 setting.

First, we target 2-key 4-EM in the Q1 setting, improving for the first time over the $T = N$ asymptotic complexity given by Grover's exhaustive search of the key. The time-memory trade-offs of the new attacks are given in Table 2,

Table 2. Quantum attacks on 2-key 4-EM. In this table, T and D are the number of queries to the round permutations and to the black-box cipher, respectively.

Adversary	Time-data trade-off	Optimal	Memory	Reference
Q2		$\begin{cases} T = \sqrt{N} \log N \\ D = \log N \end{cases}$	$\mathsf{polylog}(N)$	Offline-Simon [3]
Q1		$\begin{cases} T = \sqrt{N} \log N \\ D = N \end{cases}$	N classical	Offline-Simon [3]
Q1		$\begin{cases} T = N/2 \\ D = 1 \end{cases}$	Polynomial	Multibridge [3]
Q1 (KP)		$\begin{cases} T = N \\ D = 2 \end{cases}$	Polynomial	Grover's search
Q1 (KP)	$ND^{-1/2} + N^{1/2}D$	$\begin{cases} T = N^{5/6} \\ D = N^{1/3} \end{cases}$	D QRAQM	Section 3.2
Q1 (KP)	$ND^{-1/6} + N^{1/2}D$	$\begin{cases} T = N^{13/14} \\ D = N^{3/7} \end{cases}$	D classical + $D^{1/3}$ qubits	Section 3.3
Q1 (KP)	$D + \frac{\sqrt{N}}{\sqrt{D}} \times N^{2/3}$	$\begin{cases} T = N^{7/9} \\ D = N^{7/9} \end{cases}$	D QRACM + $N^{2/3}$ QRAQM	Section 4.2

and represented in Figure 3. While these three attacks all use classical known-plaintext queries only, they differ in their quantum memory models. Indeed, in the quantum setting, we can either allow or forbid certain types of quantum (or classical) memory with quantum random access (broadly known as qRAM, and defined in Section 2). The strongest model, quantum memory with quantum random access (QRAQM), is required by some of our algorithms.

- The first attack (Section 3.2) is a collision-based attack which uses QRAQM, and obtains the best time-memory trade-off until it reaches the complexity $T = N^{5/6}$.
- The second attack (Section 3.3) is also collision-based. While it has a worse trade-off than the first one, it does not use quantum RAM. Instead, it uses an exponential number of qubits without quantum random access.
- The third attack (Section 4.2) is our new quantum multibridge attack. It continues the time-data trade-off curve in the QRAQM setting, reaching the time complexity $T = N^{7/9}$ with $D = N^{7/9}$ classical queries. Surprisingly, as shown in Figure 3, there is a discontinuity between our attacks, and in the regime $D \in [N^{1/3}; N^{2/3}]$ the time complexity that we obtained so far is constant.

In addition to the 2-key 4-EM which alternates between the two keys, Dunkelman et al. [17] classified all possible key sequences for 3 and 4 rounds (e.g., k_0, k_0, k_0, k_1, k_1, etc.) and showed that they were either vulnerable to the clas-

sical multibridge attack, or admitted more efficient attacks. We perform a similar analysis in the quantum setting. We observe that among *all possible* key schedules alternating between two independent keys, the particular 2-key 4-EM of Figure 2 is the only one for which no quantum attack of complexity less than N, in the Q1 setting, was known prior to our work. For all other sequences, we describe simple attacks based on Grover's search, quantum collision-finding or the offline-Simon algorithm [8]. This makes the natural key-alternating schedule a particularly interesting target in the quantum setting.

From a time-data trade-off perspective, our new attacks on 2-key 4-EM are inferior to the attacks on its alternatives (Grover's search, collision-finding, offline-Simon). In particular, our quantum multibridge attack achieves at best a time complexity $N^{7/9}$, while other variants can be attacked in time $N^{2/3}$. Therefore, a strict gap remains between the best attacks on 2-key 4-EM and all other key schedules. This peculiar situation – different from the classical setting – could motivate future research into cryptanalysis for this particular structure.

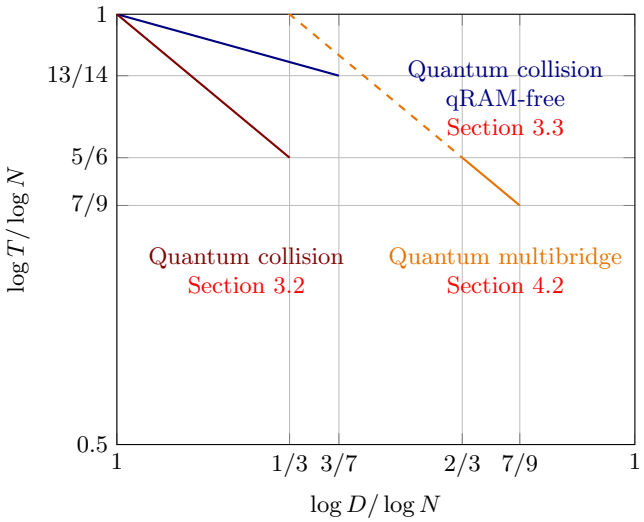

Fig. 3. Data-time trade-off of our quantum attacks on 2-key 4-EM.

Second, we target 1-key 2-EM and 2-key 6-EM using a multicollision attack adapted from [16]. On 1-key 2-EM, we obtain an attack in the Q2 setting, while on 2-key 6-EM, we get an attack in the Q1 setting using known-plaintext queries only, like our other attacks on 2-key 4-EM. In both cases these attacks improve over exhaustive key search by a factor $\sqrt{\log n}$, while classical attacks typically improve by a factor of order n. This is due to the difference in asymptotic behavior of multicollision-finding between the quantum and classical setting. Indeed, classically, finding a t-collision in an n-bit function takes time $\mathcal{O}\left(2^{n(t-1)/t}\right)$, so we can fit a value $t = \mathcal{O}(n/\log n)$ as the complexity approaches $2^n/n$. However, in the

quantum setting, the optimal algorithm [24] has complexity $\mathcal{O}\left(t2^{n\left(\frac{1}{2}-\frac{1}{2(2^t-1)}\right)}\right)$ meaning that it approaches the limit exponentially faster. This seems to be a fundamental limitation of using multicollisions in quantum cryptanalysis, which should motivate further studies of alternatives, e.g., 3-XOR algorithms.

The results of this part are summarized in Table 3.

Table 3. New quantum attacks on EM variants, based on multicollisions. Complexities are given up to constant factors.

Adversary	Time	Data	Memory	Reference
1-key 2-EM				
Q1	$2^{n/2}$	Negligible	Polynomial	Grover's search
Q2	$\frac{2^{n/2}}{\sqrt{\log n}}$	$\frac{2^{n/2}}{\sqrt{\log n}}$	$\frac{2^{n/2}}{\sqrt{\log n}}$ QRAQM	Section 5.1
2-key 6-EM				
Q1	2^n	Negligible	Polynomial	Grover's search
Q1 (KP)	$\frac{2^n}{\sqrt{\log n}}$	$\frac{2^n}{\log n}$	$\frac{2^n}{\log n}$ QRACM + $\frac{2^{n/2}}{\sqrt{\log n}}$ QRAQM	Section 5.2

While our new attacks are all in the Q1 model, making them relatively realistic, they require exponential-size QRACM, sometimes QRAQM, or a large number of qubits. This requirement is not uncommon in symmetric cryptanalysis using collision algorithms and quantum walks, but it makes the attacks strictly less practical. Yet, in the context of attacks on iterated Even-Mansour ciphers, the offline Simon's algorithm [8] previously showed that it was possible to reduce the memory footprint of existing attacks from exponential to polynomial, so we can all but wonder if this would also be possible for (some of) our new attacks.

Organization of the Paper. Section 2 introduces preliminaries of quantum computing and presents the algorithmic frameworks required for our attacks, quantum search and MNRS quantum walks. Our new attacks are detailed in the next sections. First, our two collision-based quantum attacks on 2-key 4-EM are given in Section 3. Second, our new quantum multibridge attack on 2-key 4-EM is given in Section 4. Our quantum multicollision-based attack on 2-key 6-EM is detailed in Section 5. Our classification of attacks on 2-key EM with different scheduling is given in Section 6.

2 Preliminaries

In this paper, we assume basic knowledge of quantum computing and quantum circuits such as qubits, quantum gates, amplitudes, the measurement principle, the ket notation $|\cdot\rangle$, and refer to [35] for detailed definitions.

Cost Metrics. We write quantum algorithms in the *quantum circuit model*, in which a computation is decomposed into a sequence of elementary quantum gates. One may then consider the *circuit depth*, *circuit width* and *total gate count* as relevant cost metrics. In this paper, unless stated otherwise, by *time complexity*, noted T, we mean the total gate count. In particular, we do not consider the parallelization of our algorithms, nor the trade-offs between circuit depth and width. We also count the number of qubits, and additionally, the classical memory that is used by the algorithm. In the key-recovery attacks that we consider, this gate count can be approximated by counting the number of queries to the public round permutations, while the number of queries to the black-box cipher is the data complexity D. There are sometimes other non-negligible costs, like comparisons of two n-bit values (with a linear gate count), which are balanced with an equivalent query term. This allows us to give most complexities in $\mathcal{O}$ notation, omitting constants but not linear factors.

Simon's algorithm applied on an n-bit function contains $\mathcal{O}(n)$ queries and $\mathcal{O}(n^3)$ gates to solve a Boolean linear system. We assume that the system solving time is upper bounded by the cost of queries. This allows us to simply write a multiplicative $\log N$ factor in the time complexity for a Simon-based oracle, as in Table 2 for example.

Quantum Oracle Access. Quantum symmetric cryptanalysis considers two distinct and non-equivalent scenarios based on the type of oracle access the adversary is allowed when querying the secret-key primitives.

- In the *Q1 model*, the oracle access does not differ from classical attacks, e.g., we allow classical chosen-plaintext or chosen-ciphertext queries to black-boxes.
- In the *Q2 model*, we allow the attacker to query its black-boxes as *quantum oracles*. For example, quantum chosen-plaintext queries to a block cipher E_K can be realized with the unitary operator:

$$|x\rangle |0\rangle \mapsto |x\rangle |E(x)\rangle \quad . \tag{1}$$

  This model has been used to break many symmetric constructions by embedding a hidden period problem [29,30,28,36]. Though they do not model a realistic quantum attacker (e.g., a "store now, decrypt later" adversary), Q2 attacks are interesting from a provable security perspective, since they give upper bounds on what one can prove if the primitives have quantum access.

The two models are drastically different. As an example, the one-round Even-Mansour cipher is Q2 insecure [30] but Q1 secure [1]. In both models, public components are freely accessible using quantum oracles, as they can be implemented as quantum circuits.

Quantum RAM. Many cryptanalysis algorithms require access to a large (exponential size) memory. In the quantum setting, this memory access can take

different forms, whether the memory, or the indexing, is classical or quantum. "Quantum RAM" is the common name of memory models which allow quantum indexing. More precisely, consider the following "qRAM gate", which swaps a single-bit target register y with a memory register x_i depending on an index register i:

$$|i\rangle |y\rangle |x_0, \ldots, x_{\ell-1}\rangle \xmapsto{\mathsf{qRAM}} |i\rangle |x_i\rangle |x_0, \ldots, x_{i-1}, y, x_{i+1}, x_{\ell-1}\rangle \quad . \tag{2}$$

This operation is unitary. In the standard quantum circuit model, it can be implemented with $\mathcal{O}(\ell)$ gates using a so-called "table lookup" circuit. The QRAQM (quantum random-accessible quantum memory) model considers that the qRAM gate has cost 1, i.e., it is added to the allowed elementary gate set. The QRACM (quantum random-accessible classical memory) is a weaker setting in which the memory cells $x_0, \ldots, x_{\ell-1}$ must be classical and the qRAM operation is simply an access:

$$|i\rangle |y\rangle \xmapsto{\mathsf{qRAM}} |i\rangle |y \oplus x_i\rangle \quad . \tag{3}$$

qRAM is a very strong assumption, but also an essential component of many quantum algorithms based on collision-finding [12] and quantum walks [34] which are used in this paper. In some cases, it is possible to replace QRAQM by QRACM, or by an exponential number of qubits without quantum access; both cases are equally interesting to us as they make the model strictly more realistic.

2.1 Properties of Random Functions

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a random n-bit function. For any $t \geq 0$, the average number of t-collisions of f, i.e., of tuples $(x_1, \ldots, x_t)$ of distinct entries such that $f(x_1) = \ldots = f(x_t)$, is $2^n e^{-1}/t!$ [20]. By Stirling's formula, a random function admits on average a constant number of t-collisions for $t = \mathcal{O}(n/\log n)$. In addition, the probability that it admits an n-collision vanishes. Finally, even if we consider a family of 2^{rn} random functions for a constant r, the probability that any one of them admits an n-collision vanishes.

2.2 Amplitude Amplification

Quantum Amplitude Amplification [11] (QAA) is a generalization of Grover's search which has become one of the main tools used in quantum cryptanalysis. It amplifies the probability of success of any probabilistic algorithm in which we can check the output.

Theorem 1 (From [11]). *Let $\mathcal{A}$ be a quantum algorithm without measurement that, starting from $|0\rangle$, creates a state of the form:*

$$\mathcal{A}|0\rangle = \alpha |G\rangle |1\rangle + \beta |B\rangle |0\rangle \quad , \tag{4}$$

where $|G\rangle$ and $|B\rangle$ are both normalized quantum states which are respectively a superposition of "good outcomes" and "bad outcomes", and $|\alpha|^2$ is the probability

of success. Define the QAA iterate *as:* $Q = -\mathcal{A}O_0\mathcal{A}^\dagger O$ *where* O_0 *flips the phase of any non-zero basis state, and* O *is the controlled phase flip controlled by the last ("flag") qubit. Then for any* $t \geq 0$*:*

$$Q^t\mathcal{A}\left|0\right\rangle = \sin((2t+1)\arcsin\alpha)\left|G\right\rangle\left|1\right\rangle + \cos((2t+1)\arcsin\alpha)\left|B\right\rangle\left|0\right\rangle \quad . \tag{5}$$

It follows that after $t = \left\lfloor \frac{\pi}{4}\frac{1}{\arcsin\alpha} \right\rfloor = \mathcal{O}(1/\alpha)$ iterations, a measurement will project on $\left|G\right\rangle$ with constant probability. Furthermore, the probability distribution of the different possible good outcomes remains unchanged.

If the probability of success of $\mathcal{A}$ is only known up to a constant factor, we can just take a guess at the appropriate number of iterations, and still succeed with constant probability.

Quantum Collision Search. The BHT algorithm [12] is a quantum collision search algorithm, which more generally solves a *multi-target preimage* problem. Consider a function $f : \{0,1\}^n \to \{0,1\}^n$ and a list of targets L of size 2^t. The goal is to find x such that $f(x) \in L$.

If QRACM is available, we can first construct the memory containing the list L, in time 2^t, then run a quantum search for x. Testing if x is good requires to implement the *membership* unitary: $\left|y\right\rangle\left|0\right\rangle \mapsto \left|y\right\rangle\left|y \in L\right\rangle$. With QRACM, this can be done in polynomial time. The total gate count is therefore: $\widetilde{\mathcal{O}}\left(2^t + 2^{n/2}/2^{t/2}\right)$. This can be used for collision search by simply taking arbitrary targets and optimizing $t = n/3$, giving an optimal time $\widetilde{\mathcal{O}}\left(2^{n/3}\right)$ for random functions.

The algorithm of [13] gives a trade-off when QRACM is not available, and uses only an exponential-size classical memory with sequential access. It implements the membership using elementary quantum gates (similarly to a "table lookup" circuit). To compensate for this cost, one takes a sub-list L' of *distinguished points*, which are point x such that $f(x)$ has an arbitrary fixed $2t/3$-bit prefix.

In the first step, one constructs a list of distinguished points. In the collision search case, a distinguished point can be found in time $\mathcal{O}\left(2^{t/3}\right)$ using quantum search. We create a list L' of size $\mathcal{O}\left(2^{t/3}\right)$, which can be constructed in time $\mathcal{O}\left(2^{2t/3}\right)$. In the multi-target preimage case, the list L' is only constructed in time $\mathcal{O}(2^t)$, by filtering the initial list of 2^t targets.

In the second step, one performs a two-level QAA. The analysis of this algorithm is actually quite easy thanks to Theorem 1:

- First, one constructs an algorithm that produces a superposition of distinguished points using a quantum search. Using $2^{t/3}$ iterations, this algorithm succeeds with constant probability, i.e., produces a superposition of points which are distinguished with constant probability.
- Then, one appends to this algorithm another step which tests whether the point belongs to the sub-list L'. This test costs $\widetilde{\mathcal{O}}\left(2^{t/3}\right)$ quantum gates. The composition of these algorithms has a probability of success $\mathcal{O}\left(2^{t/3} \times 2^{2t/3-n}\right)$, because the first step has constant probability of sampling a distinguished point, and once this is done, we will have a probability $\mathcal{O}\left(2^{t/3} \times 2^{2t/3-n}\right)$ to match one of the distinguished points in L'.

- Finally, by Theorem 1, by making $\mathcal{O}\left(2^{(n-t)/2}\right)$ iterations of quantum search, we succeed with constant probability.

In the multi-target preimage case, the total gate count is:

$$\widetilde{\mathcal{O}}\left(2^t + 2^{(n-t)/2} \times 2^{t/3}\right) = \widetilde{\mathcal{O}}\left(2^t + 2^{n/2-t/6}\right) \quad . \tag{6}$$

It is minimized for $t = 3n/7$ at $\mathcal{O}\left(2^{3n/7}\right)$.

Multicollision Search. In the classical setting, a t-collision of a random function with n-bit range and domain can be found in time and memory $\mathcal{O}\left(2^{n(t-1)/t}\right)$.

In the quantum setting, Hosoyamada et al. [24] give a t-collision-finding algorithm of complexity $\mathcal{O}\left(t2^{n\frac{2^{t-1}-1}{2^t-1}}\right)$ on an n-bit random function, where the constant in the $\mathcal{O}$ does not depend on t. This algorithm uses $\mathcal{O}\left(2^{n\frac{2^{t-1}-1}{2^t-1}}\right)$ QRACM and constructs successive lists of 2-collisions, 3-collisions, etc. This has been proven to be optimal by Liu and Zhandry [33] for a random function.

2.3 Search via Quantum Walk

Let G be a regular undirected graph containing a proportion ε of *marked* vertices, which satisfy a given property. The MNRS quantum walk framework [34] generically finds a marked vertex by performing the equivalent of a classical random walk. It relies on the following unitaries: an *update* unitary U (of cost U) which, given a vertex, produces a uniform superposition of its neighbors; a *setup* unitary S (cost S) which produces a uniform superposition of vertices; a *check* unitary C (of cost C) which finds whether a vertex is marked. The latter is most often made negligible by coding the "marked" information as a flag into the vertex data structure itself.

Let δ be the *spectral gap* of G; in particular, $\frac{1}{\delta}$ is the number of steps for a random walk on G to converge to a uniform distribution of vertices. We have the following.

Theorem 2 ([34]). *There exists a quantum algorithm that finds a marked vertex in time:*

$$\widetilde{\mathcal{O}}\left(S + \frac{1}{\sqrt{\varepsilon}}\left(C + \frac{1}{\sqrt{\delta}}U\right)\right) \quad . \tag{7}$$

On a high level, this algorithm can be considered as a variant of QAA in which there is a trade-off between the cost to setup the uniform superposition of marked vertices (S) and the cost to reflect through it $\left(\frac{1}{\sqrt{\delta}}U\right)$. In the latter, one notices that there is a quadratic speedup compared to the cost of resampling a random vertex in a classical random walk $\left(\frac{1}{\delta}U\right)$.

Most applications of MNRS quantum walks, especially in cryptography, use *Johnson graphs*. A Johnson graph $J(X, R)$ is the graph of subsets of size R of a set X of size N, where two vertices are connected iff they differ by a single

element. The graph has $\binom{N}{R}$ vertices and its spectral gap is $\delta = \frac{N}{R(N-R)} \simeq \frac{1}{R}$ when $R \ll N$. In addition, the spectral gap of a *product Johnson graph* $J(X,R)^r$ satisfies $\delta \geq \frac{1}{r}\frac{N}{R(N-R)}$ [27].

In order to be efficient in gate count, the MNRS algorithm requires efficient operations on sets which can be implemented using QRAQM. To be more precise, a vertex is identified with a *data structure* that represents at least the subset of elements that it contains, with additional information, such as other sets, and also the "marked" flag. All this information, including the flag, is updated upon updating the vertex (i.e., replacing an element). This data structure is unique to a vertex. It is especially important that it can be updated in polynomial time at each walk step, in a *history-independent* way, so that there is a one-to-one correspondence between vertices and their data structures.

Previous works like [26,7] detail suitable data structures for sets with additional information. These are *quantum augmented radix trees*, which represent a set of elements using a radix tree, with additional information attached to each node (if required). Quantum radix trees allow for fast insertion, deletion, and retrieving partial collisions with a given input. In this paper, we will use them in a black-box manner: a node in the graph will have a data structure constructed from multiple quantum radix trees.

3 Quantum Collision Attack on 2-key 4-EM

In this section, we consider the 4-round Even-Mansour with two alternating keys and 4 independent permutations, as shown in Figure 2.

The idea of our quantum attack is the following. Suppose that we have access to many known plaintext-ciphertext pairs for the full 2-key 4-EM cipher. We now guess the key k_0. As shown in Figure 4, this reduces the cipher to a single-round Even-Mansour with one key (1-key 1-EM), where the permutation Π' is $\Pi'(x) := \Pi_3(\Pi_2(x) \oplus k_0)$. We call this the *inner Even-Mansour* cipher.

Known plaintext-ciphertext pairs for the 2-key 4-EM can be processed into known plaintext-ciphertext pairs for the inner Even-Mansour. Thus, we can attack the 2-key 4-EM using classical known plaintexts only.

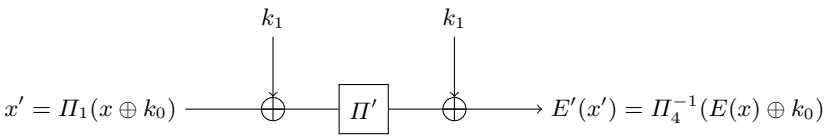

Fig. 4. Inner Even-Mansour cipher.

In the following, we first recall the classical (and quantum) *Slidex* attack on the single-round Even-Mansour, which we then use to attack the full cipher.

3.1 Classical and Quantum Slidex

We start by describing the classical Slidex attack on 1-key 1-EM, i.e., a single-round Even-Mansour cipher with a single key. The attack is given in [17, Section 3.1] and is an adaptation of the Slidex attack for 2-key 1-EM from [18].

Let $E'(x) = k_1 \oplus \Pi'(x \oplus k_1)$ be the cipher to attack. We are given D known plaintext-ciphertext pairs $(x, E'(x))$, and we compute the list of pairs $(x, x \oplus E'(x))$. Then, the goal is to find Δ such that: $\Pi'(\Delta) \oplus \Delta = x \oplus E'(x)$. Indeed, this implies:

$$\Pi'(\Delta) \oplus \Delta = x \oplus k_1 \oplus \Pi'(x \oplus k_1) \ . \tag{8}$$

Such a collision gives a candidate for the key: $x \oplus \Delta$, which is the right one with constant probability.

In the classical setting, with D known plaintext-ciphertext pairs, we construct the list in time D, we then need to test N/D values of Δ in order to find the right key with high constant probability, yielding a time complexity of $\max(D, N/D)$.

In the quantum setting, let $f : \Delta \mapsto \Pi'(\Delta) \oplus \Delta$. The goal is to find a pair (Δ, x) such that $f(\Delta) = x \oplus E'(x)$, i.e., to solve a multi-target preimage problem on f. The list of targets is of size D, given by the data, and the domain of f is of size N, given by the possible choices of Δ. In addition, f is close to a random function since Π' is a random permutation. We can solve this problem in different ways depending on the memory model:

- With QRACM, we can use BHT collision search [12]. The first step of the algorithm constructs the QRACM with D queries, and has complexity $\mathcal{O}(D)$. The second step performs a Grover search for Δ such that $f(\Delta)$ belongs to the table. This search has $\mathcal{O}\left(\sqrt{N/D}\right)$ iterations. The membership test is done efficiently thanks to QRACM access. The total complexity is $T = \mathcal{O}\left(D + \sqrt{N/D}\right)$, which is minimal at $D = N^{1/3}$. This is the trade-off given in [30].
- With classical memory only, we can use the algorithm of [13]. The first step constructs a classical (not qRAM) memory with $D^{1/3}$ elements by filtering the known-plaintext queries. The second step performs a quantum search. The total complexity is $T = \mathcal{O}\left(D + N^{1/2}/D^{1/6}\right)$, which is minimal at $D = T = N^{3/7}$. This is a trade-off given in [23].

3.2 Quantum Attack with QRAQM

Our first attack performs a quantum search on k_0, using the BHT variant of the Slidex attack on the inner Even-Mansour cipher. It is given in Algorithm 1.

Algorithm 1 contains two nested QAAs. We analyze them using Theorem 1.

If k_0 is the good key, a random Δ will lead to the right key guess (k_0, k_1) with probability D/N, so the second QAA at Steps 5-10 produces a superposition where the flag bit is 1 with constant probability. If k_0 is the wrong key, this second QAA produces a superposition where the flag bit is never 1 (the only case when it is written 1 is the case where k_0, k_1 is the secret key).

Algorithm 1 Quantum collision attack with QRAQM on 2-key 4-EM

Input: D known-plaintext queries $(x, E(x))$
Output: key candidate k_0, k_1

1: **amplify with** $\sqrt{N}$ **iterations**
2: Start in the state $|0\rangle$
3: Create a uniform superposition of $k_0 \in \{0,1\}^n$
▷ Classically: sample k_0 uniformly at random
4: From each plaintext-ciphertext $(x, E(x))$ compute the pair $(x', E'(x')) = (\Pi_1(x \oplus k_0), \Pi_4^{-1}(E(x) \oplus k_0))$
▷ This takes time D, and the pairs $(x', E'(x'))$ need to be stored in QRAQM.
5: **amplify with** $\sqrt{N/D}$ **iterations**
6: Create a uniform superposition of $\Delta \in \{0,1\}^n$
▷ Classically: sample Δ uniformly at random
7: Find x' in the list of pairs such that $\Pi'(\Delta) \oplus \Delta = x' \oplus E'(x')$, where $\Pi'(y) := \Pi_3(\Pi_2(y) \oplus k_0)$
▷ To implement efficiently this step, we need QRAQM access
8: If such an x' exists, set $k_1 := \Delta \oplus x'$
9: Test the key candidate (k_0, k_1) using three plaintext-ciphertext pairs
10: **Good outcome:** the candidate (k_0, k_1) is correct
11: **end amplify**
▷ At this point, we have created a quantum state that contains a superposition of k_0, Δ, k_1 and corresponding data. In this QAA, we simply reuse the flag written at Step 9
12: **Good outcome:** same as Step 10
13: **end amplify**

Therefore, the entire amplified algorithm from Steps 2-10 produces a superposition over k_0, Δ, k_1 and corresponding flag bit where the flag is 1 with probability $\mathcal{O}(1/N)$. By Theorem 1, after $\mathcal{O}\left(N^{1/2}\right)$ iterations, we measure the good key with constant probability. The total gate count is:

$$\mathcal{O}\left(N^{1/2}\left(\sqrt{N/D} + D\right)\right) = \mathcal{O}\left(N/D^{1/2} + N^{1/2}D\right) \tag{9}$$

which is minimized at $\mathcal{O}\left(N^{5/6}\right)$ for $D = N^{1/3}$.

Regarding the memory complexity, we need a classical memory of size D to store the initial plaintext-ciphertext pairs. During the amplified algorithm, under the current value of k_0 (which is in superposition), we need to recompute the list of pairs $(x', E'(x))$ and store it in quantum-accessible memory. Since this new list depends on k_0, it is also quantum, which is why the algorithm requires a QRAQM of size D.

Remark on Optimality. Within the inner loop of Algorithm 1, we are essentially solving a claw-finding problem between two lists of sizes N/D and D. Using the best available algorithms, quantum claw-finding between two lists of size M and M' can be solved in time $(MM')^{1/3}$ if $M \leq M' \leq M^2$ and $(M')^{1/2}$ if $M' \geq M^2$ [39].

We can distinguish three cases. • if $D \geq (N/D)^2 \iff D \geq N^{2/3}$, then the complexity is in $\sqrt{N} \times \sqrt{D} \geq N^{5/6}$, which does not do better than our algorithm. • if $(N/D) \geq D^2 \iff D \leq N^{1/3}$, then the complexity is $\sqrt{N} \times \sqrt{N/D}$ which is exactly the time-data trade-off curve of our algorithm. • in all other cases, the best achievable complexity is $\sqrt{N} \times (N/D \times D)^{1/3} = N^{5/6}$, which is already the best trade-off point achieved at $D = N^{1/3}$.

In other words, Algorithm 1 achieves the best trade-off possible if we only use inner collision search, regardless of memory complexity.

3.3 Quantum Attack Without qRAM

In our second attack, we modify the amplified algorithm by using the multi-target preimage search of [13] instead of the BHT one (see Section 2.2). The new algorithm is Algorithm 2.

Algorithm 2 Quantum collision attack without qRAM on 2-key 4-EM

Input: D known-plaintext queries $(x, E(x))$
Output: key candidate k_0, k_1

1: **amplify with** $\sqrt{N}$ **iterations**
2: Start in the state $|0\rangle$
3: Create a uniform superposition of $k_0 \in \{0,1\}^n$
▷ Classically: sample k_0 uniformly at random
4: From each plaintext-ciphertext $(x, E(x))$ compute the pair $(x', E'(x')) = (\Pi_1(x \oplus k_0), \Pi_4^{-1}(E(x) \oplus k_0))$. Select only the pairs such that $E'(x') \oplus x'$ starts with $d = \frac{2}{3}\log_2 D$ zeroes and write them in a memory of size $D^{1/3}$
▷ This takes time D, but only $D^{1/3}$ memory, using a sequential circuit. The zero-prefix is an arbitrary choice.
5: **amplify with** $\sqrt{N/D}$ **iterations**
6: **amplify with** $D^{1/3}$ **iterations**
7: Create a uniform superposition of $\Delta \in \{0,1\}^n$
▷ Classically: sample Δ uniformly at random
8: Compute $\Pi'(\Delta) \oplus \Delta$ where $\Pi'(x) := \Pi_3(\Pi_2(x) \oplus k_0)$
9: **Good outcome:** $\Pi'(\Delta) \oplus \Delta$ starts with d zeroes
10: **end amplify**
11: Find x' in the memory such that $\Pi'(\Delta) \oplus \Delta = x' \oplus E'(x')$
▷ This steps now takes time $D^{1/3}$, since the list has size $D^{1/3}$ and we do not use qRAM
12: If such an x' exists, set $k_1 := \Delta \oplus x'$
13: Test the key candidate (k_0, k_1) using three plaintext-ciphertext pairs
14: **Good outcome:** the candidate (k_0, k_1) is correct
15: **end amplify**
▷ At this point, we have created a quantum state that contains a superposition of k_0, Δ, k_1 and corresponding data. In this QAA, we simply reuse the flag written at Step 13
16: **Good outcome:** same as Step 13
17: **end amplify**

The only difference between Algorithm 2 and Algorithm 1 lies in the collision step, where we use a smaller list and try to find a *distinguished* value of Δ. First, the QAA at steps 6-10 has constant success probability after $D^{1/3}$ iterations (since a random Δ satisfies the prefix condition with probability $D^{-2/3}$). Next, the probability that a given Δ is the correct one (and leads to the key) is D/N, whether or not it satisfies the prefix condition. Therefore the algorithm at Steps 6-14 has success probability $\mathcal{O}(D/N)$, when k_0 is the correct key, and 0 otherwise. This justifies to take $\sqrt{N/D}$ iterations at Step 5, and like before, $\sqrt{N}$ at Step 1.

While we use D *classical* memory to store the known-plaintext queries, the list used during multi-target preimage search is only of size $D^{1/3}$. Since this list depends on the current choice of k_0, it needs to be stored in quantum memory (i.e., qubits). However, we do not access it quantumly; the algorithm does not use qRAM.

The time complexity becomes:

$$\mathcal{O}\left(N^{1/2}\left(D + N^{1/2}/D^{1/6}\right)\right) \quad , \tag{10}$$

which is minimal at $N^{13/14}$, using $N^{3/7}$ classical memory and $N^{1/7}$ qubits.

4 Quantum Multibridge Attack on 2-Key 4-EM

The attacks from Section 3 improve over Grover's exhaustive key search if sufficient memory, either QRAQM or qubits, is available. However, even when increasing the data complexity, the improvement over Grover search will eventually reach a limit. The same can be seen in classical attacks on 2-key 4-EM which would use the same collision strategy: the time-data trade-off follows $T = N^2/D$ until $D = N^{1/2}$, after which the attack on the inner 1-key 1-EM cipher cannot make use of more data.

In [17], Dinur et al. describe an improved attack on 2-Key 4-EM, named the *multibridge* attack. In [3], Anand et al. describe a quantum version of this attack, but without improving by more than a constant factor compared to Grover's exhaustive key search. In this section, we devise a new quantum multibridge attack, based on a quantum walk. This attack uses QRAQM. Contrary to the collision-based attack of Section 3, it does not have a qRAM-free variant. The intuitive reason for this is that the quantum multibridge performs well in a high data complexity regime, which is contradictory with the regime of low memory complexity which qRAM-free attacks tend to favor.

4.1 Description of the Classical Attack

We start by giving a description of the multibridge attack as a *list merging* algorithm. This description differs slightly from [17], but reuses similar notations.

Using the notation of Figure 5, consider the following situation. We fix a constant Δ, and we encrypt a plaintext x through the 2-key 4-EM cipher. We

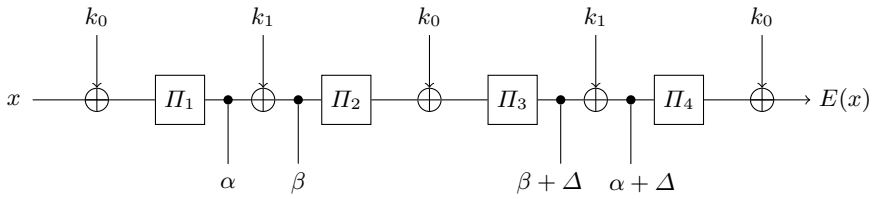

Fig. 5. Internal collision event in a 2-key 4-EM cipher.

note $\alpha := \Pi_1(k_0 \oplus x)$ and $\beta = k_1 \oplus \alpha$. With probability N^{-1} for a given x, the following collision event (E) will occur:

$$(E) \quad \beta \oplus \Delta = \Pi_3(k_0 \oplus \Pi_2(\beta)) \ .$$

This is represented in Figure 5. If this event occurs, the following system of equations is then satisfied:

$$\begin{cases} x \oplus E(x) \oplus \Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta) = 0 \\ x \oplus \Pi_1^{-1}(\alpha) \oplus \Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta) = 0 \end{cases} \tag{11}$$

Suppose that we start from D known-plaintext queries $(x, E(x))$. It follows that with probability D/N, there exists x such that the event (E) occurs. After trying N/D different values for Δ, this event will eventually occur with constant probability.

The goal of the classical multibridge attack is to identify candidate quadruples Δ, x, α, β, and to deduce candidates for (k_0, k_1) from them. This is summarized in Algorithm 3.

Algorithm 3 Classical multibridge attack.

Input: D known-plaintext queries $(x, E(x))$
Output: key candidate k_0, k_1

1: Choose an arbitrary set $X \in \{0,1\}^n$ of size N/D
2: **for** $\Delta \in X$ **do**
3: Find all x, α, β such that:

$$(x \oplus E(x), x) \oplus (\Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta), \Pi_1^{-1}(\alpha)) \\ \oplus (0, \Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta)) = (0,0)$$

4: For all solutions (expected $D \times N^2 \times N^{-2} = D$) deduce $k_0 = x \oplus \Pi_1^{-1}(\alpha)$ and $k_1 = \alpha \oplus \beta$
5: **Return** k_0, k_1 if they match a few plaintext-ciphertext pairs
6: **end for**

Therefore, the main step in the algorithm is to solve a 3-XOR problem between three lists of pairs of bit-strings, i.e., find a triple in the lists which XORs to 0:

- The list L_x of all $(x \oplus E(x), x)$, of size D, built from the data and indexed by x;
- The list L_α of all $(\Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta), \Pi_1^{-1}(\alpha))$, indexed by α, of size N;
- The list L_β of all $(0, \Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta))$, indexed by β, of size N.

This 3-XOR instance can be solved in time N and memory D, as one can do the following:

- Merge the list of x and the list of α using the first condition of Equation 11, i.e., construct the list of all (x, α) such that $\Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta) = x \oplus E(x)$. For this one stores L_x in memory and exhaustively searches through L_α, by iterating over the values of α in the set $\{0,1\}^n$, computing the element of L_α, and checking if it belongs to L_x. The output list is of expected size $D \times N/N = D$;
- Merge the list of (x, α) and the list of β using the second condition of Equation 11. Again the β do not need to be stored in memory for this, and the output list is of expected size D.

Therefore, the time complexity is $T = N^2/D$ for any D, and the memory is D.

4.2 Quantum Attack based on a Quantum Walk

Like the classical attack, the quantum attack starts by querying D arbitrary plaintexts. We denote by L_x the list of $(x, E(x) \oplus x)$, which is of size D, and is now stored in QRACM, allowing fast quantum access in the remainder of the algorithm. We also select an arbitrary subset $X \subseteq \{0,1\}^n$ of size N/D, for example using a prefix.

Then, we perform QAA over an algorithm that:

- Samples a random $\Delta \in X$;
- Finds if this value of Δ leads to the good key (k_0, k_1), by searching for key candidates using a quantum walk.

We expect that there exists a single good Δ leading to the key, and we just need to ensure that the internal algorithm, for this good Δ, produces the good key with constant probability (for all other Δ, there is no solution). This ensures that after $N^{1/2}/D^{1/2}$ iterations of QAA, we find the key with constant probability.

Finding the Key at Fixed Δ. Suppose now that Δ is fixed. We solve the 3-XOR problem outlined above using an MNRS quantum walk algorithm.

The graph of the walk is a product Johnson graph, where each node is built from a subset of fixed size R of $\{0,1\}^n$ for α and a subset of size R for β. The vertex data structure is defined from these subsets as follows:

- Construct the list L'_α of all $(\Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta), \Pi_1^{-1}(\alpha), \alpha)$, where α belongs to the vertex' subset. This list is stored as a set of bit-strings in a quantum radix tree. It is a sub-list of L_α previously considered in the classical attack.
- Construct the list L'_β of all $(\Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta), \beta)$, where β belongs to the vertex' subset. This list is stored as a set of bit-strings in a quantum radix tree. It is a sub-list of L_α previously considered in the classical attack.
- Take the list $L_{x\alpha} \subseteq L_x \times L_\alpha$ of all $(x \oplus \Pi_1^{-1}(\alpha), \alpha, x)$ such that $E(x) \oplus x = \Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta)$. This list is stored as a set of bit-strings in a quantum radix tree.
- Take the list $L_{x\alpha\beta} \subseteq L_{x\alpha} \times L_\beta$ of all (α, x, β) such that $x \oplus \Pi_1^{-1}(\alpha) \oplus (\Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta)) = 0$. This list is stored as a set of bit-strings in a quantum radix tree.
- In the list $L_{x\alpha\beta}$, if a candidate x, α, β leads to the good key (k_0, k_1), flag the vertex as "marked". Otherwise the vertex is unmarked.

Our goal is to find a marked vertex, i.e., a vertex which contains a choice of α, β leading to the key. For the good Δ, we expect a single such choice to exist (corresponding to a single internal collision event, for a single input x). Therefore, the proportion of marked vertices is: $\varepsilon = \left(\frac{R}{N}\right)^2$. For $N \gg R$, given that we have a product Johnson graph, the spectral gap is $\delta = \frac{1}{R}$ [27].

Cost of Operators. The checking operator is trivial, since the vertex data structure contains the flag that indicates if the vertex is marked, so $C = 1$.

For the update operator, we must ensure that the vertex data structure can be updated in polynomial time when inserting or deleting a value of α and / or β. If this can be done, then the setup operator can be efficiently implemented by a sequence of updates as well. For inserting α, we perform the operations given in Algorithm 4. The operations to delete, and the case of β, are similar. They have the same cost since we operate on the same number of elements.

Algorithm 4 Update algorithm

Input: value α to add, current vertex data structure

1: Compute $(\Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta), \Pi_1^{-1}(\alpha), \alpha)$ and insert this value in L_α
2: Find all $x \in L_x$ such that $E(x) \oplus x = \Pi_1^{-1}(\alpha) \oplus \Pi_4(\alpha \oplus \Delta)$
3: For all such x, insert $(\Pi_1^{-1}(\alpha), \alpha, x)$ in $L_{x\alpha}$
4: For all such x, find all $\beta \in L_\beta$ such that $x \oplus \Pi_1^{-1}(\alpha) \oplus (\Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta)) = 0$
5: Insert each (α, x, β) into $L_{x\alpha\beta}$

To guarantee that Algorithm 4 runs in polynomial time, we verify that:

- For any α, Δ, the number of elements x found at Step 2 is globally bounded by a polynomial. Indeed, finding t values here would mean that the function $x \mapsto E(x) \oplus x$, which should behave as random, admits a t-collision. Since

the probability of a random function to have an n-collision is negligible, we now that we should always have at most n elements here to update, and we can do this with a fixed-size circuit.

- For any α, Δ, x, the number of elements β found at Step 4 is globally bounded by a polynomial. Indeed, we found only n valid x at most at Step 2. Then, for each x, we expect at most n valid values of β. Otherwise, we would have an n-collision in the function $\beta \mapsto \Pi_2(\beta) \oplus \Pi_3^{-1}(\beta \oplus \Delta)$, which happens only with negligible probability (even for all possible Δ).

Algorithm. In summary, when we run the quantum walk, we assume that a solution exists: if Δ is good, marked vertices will appear with constant probability in the output, and any marked vertex contains a key candidate. Otherwise, there will be no marked vertices in the output.

By Theorem 2, the quantum time complexity of the walk, up to polynomial factors, is the following:

$$R + \frac{1}{\sqrt{\epsilon}} \frac{1}{\sqrt{\delta}} = R + \frac{N}{R}\sqrt{R} = R + \frac{N}{\sqrt{R}} \ . \tag{12}$$

This is optimal if we use $R = N^{2/3}$, giving a time $N^{2/3}$. This corresponds to the time complexity of a quantum walk for claw-finding when we look for a single claw (a variant of Ambainis' algorithm). That is, we are looking for a unique pair $(\alpha, \beta) \in \{0,1\}^2$ satisfying some relation.

The time complexity of the full attack takes into account the search over Δ, and the construction of the QRACM storing the list of $(x, E(x))$. We obtain:

$$D + \underbrace{\sqrt{\frac{N}{D}}}_{\text{Quantum search on } \Delta} \qquad N^{2/3} = D + N^{7/6} D^{-1/2} \ . \tag{13}$$

where the term D is due to constructing the QRACM where we store the list of $(x, E(x))$. Interestingly, this attack is only better than Grover's search (complexity N) for D such that:

$$N^{7/6} D^{-1/2} \leq N \iff N^{1/6} \leq D^{1/2} \iff D \geq N^{1/3} \ . \tag{14}$$

The optimal time complexity is reached at:

$$D = N^{7/6} D^{-1/2} \iff T = D = N^{7/9} \ . \tag{15}$$

The memory used is $N^{7/9}$ QRACM (for the data) and $N^{2/3}$ QRAQM (for the set data structures in the MNRS walk).

In Section 3.2, we had an attack which worked only until $D = N^{1/3}$, with a minimal complexity $N^{5/6}$. This means that the quantum multibridge attack is not useful for a certain range of D. In this range, which starts from $D = N^{1/3}$ and until we have $N^{7/6}/D^{1/2} = N^{5/6} \implies D = N^{2/3}$, the best known complexity displays a plateau at $T = N^{5/6}$.

These different regimes are represented in Figure 3.

5 Quantum Multicollision Attack on 1-key 2-EM and 2-key 6-EM

In this section, we give additional attacks on EM variants, starting with a one-key variant with two rounds (1-key 2-EM), which we then use to attack a 2-key variant with 6 rounds (2-key 6-EM). Although these attacks offer a very limited speedup, they are the first full key-recovery attacks reaching asymptotically below Grover's exhaustive key search (respectively $2^{n/2}$ and 2^n quantum time) for these variants[1].

5.1 Q2 Attack on 1-key 2-EM

The 1-key 2-EM variant is represented in Figure 6. Contrary to Section 3 and Section 4, here we will consider a Q2 adversary, capable of making superposition queries.

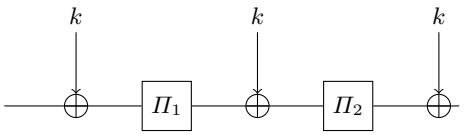

Fig. 6. Two-round iterated Even-Mansour cipher (1-key 2-EM).

Classical Multicollision Attack. The classical attack that we will follow is originally from [16]. More advanced attacks on two-round Even-Mansour were given later, including improved time-memory trade-offs [25,32]. However, we did not find a way to exploit these attacks further.

Let $F : \{0,1\}^n$ be the function defined by $F(x) = x \oplus \Pi_1(x)$. When Π_1 is a random permutation, F behaves closely to a random function, and it will be considered as such in the attack. With this definition, one can rewrite the cipher E as:

$$\begin{aligned} E(x) = k \oplus \Pi_2(k \oplus \Pi_1(k \oplus x)) &= k \oplus \Pi_2(x \oplus k \oplus x \oplus \Pi_1(k \oplus x)) \\ &= k \oplus \Pi_2(x \oplus F(k \oplus x)) \ . \end{aligned} \quad (16)$$

The attack runs in two steps.

1. Offline step: one chooses a parameter t and finds a t-collision of F, i.e., a tuple of t distinct values $(x_1, \ldots, x_t)$ such that $F(x_1) = \ldots = F(x_t) = v$ where $v \in \{0,1\}^n$ is now known. For $t = \mathcal{O}(n/\log n)$ (where the constant is smaller than 1), a t-multicollision can be found in time $\mathcal{O}(2^n \log n/n)$.

[1] A distinguishing attack on 2-EM with query complexity $2^{3n/5}$ is given in [5], but it is inefficient in time.

2. Online step: assume that $\mathcal{O}(2^n \log n/n)$ known-plaintext queries are given. With high probability, one of the inputs x is such that $F(x \oplus k) = v$. This condition is equivalent to:

$$E(x) = k \oplus \Pi_2(x \oplus F(k \oplus x)) = k \oplus \Pi_2(x \oplus v) \ . \tag{17}$$

   So, for all x, we compute a key candidate $k^* := E(x) \oplus \Pi_2(x \oplus v)$. We test this key candidate using a known plaintext-ciphertext pair.

Both steps have equal complexities when $t = \mathcal{O}(n/\log n)$, giving an attack in time $\mathcal{O}(2^n \log n/n)$, memory $\mathcal{O}(2^n \log n/n)$ (due to the multicollision-finding step) and data $\mathcal{O}(2^n \log n/n)$.

The quantum attack reproduces both steps, but needs to optimize t differently.

Quantum Attack: Offline Step. In the offline step, we use the multicollision-finding algorithm of Hosoyamada et al. [24], of complexity (up to a constant):

$$t2^{n\frac{2^{t-1}-1}{2^t-1}} = t2^{n\left(\frac{1}{2}-\frac{1}{2(2^t-1)}\right)} \ . \tag{18}$$

Quantum Attack: Online Step. In the online step, we sieve the key candidates using a quantum search. In detail, we use quantum search over the following algorithm:

- Pick x uniformly at random;
- Compute $k^* := E(x) \oplus \Pi_2(x \oplus v)$;
- Check if k^* is the correct key using a known plaintext-ciphertext pair.

At the second step, we see that it is necessary to query E in superposition, which is why this attack requires Q2 queries. The entire second step uses only polynomial memory, like in the classical setting, and has a time (and query) complexity $\mathcal{O}(\sqrt{2^n/t})$.

Let us take for example $t = \frac{1}{2}\log_2 n$, then the multicollision-finding step has complexity:

$$t2^{n\left(\frac{1}{2}-\frac{1}{2(2^t-1)}\right)} \leq t2^{n\left(\frac{1}{2}-\frac{1}{2^{t+1}}\right)} = \frac{1}{2}\log_2 n 2^{n/2} 2^{-\frac{n}{2n^{1/2}}} = o\left(\frac{2^{n/2}}{\sqrt{\log n}}\right) \ , \tag{19}$$

while the search step has complexity $\mathcal{O}\left(\frac{2^{n/2}}{\sqrt{\log n}}\right)$. Contrary to classical attacks, the maximal size of a multicollision is not a limiting factor, but rather, the complexity of finding it, as there is no advantage in using t above $\mathcal{O}(\log n)$.

5.2 Q1 Attack on 2-key 6-EM

Our attack on 2-key 6-EM leverages the multicollision attack on 1-key 2-EM given above. Intuitively, by guessing k_0 (with a quantum search) the cipher is reduced to a 1-key 2-EM instance with permutations that depend on k_0. We

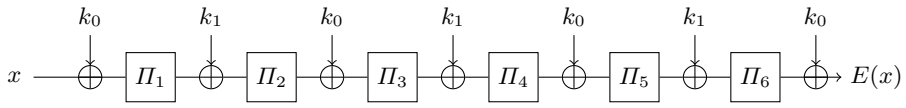

Fig. 7. Two-key iterated Even-Mansour cipher with 6 rounds.

would be able to run the attack in time $\mathcal{O}\big(2^{n/2} \times \frac{2^{n/2}}{\sqrt{\log n}}\big)$, and quantum memory $\mathcal{O}\big(\frac{2^{n/2}}{\sqrt{\log n}}\big)$.

However, we can do a better attack that requires $\frac{2^n}{\log n}$ classical queries only, which we detail in Algorithm 5.

Algorithm 5 Multicollision attack on 2-key 6-EM

Input: $\frac{2^n}{\log n}$ classical known-plaintext queries $(x, E(x))$
Output: Key candidate k_0, k_1

1: Choose a parameter $t = \mathcal{O}(\log n)$ (for example $t = \frac{1}{2}\log_2 n$)
2: **amplify with** $2^{n/2}$ **iterations**
3: Pick k_0 uniformly at random
4: Define $\Pi_1' := \Pi_3(k_0 \oplus \Pi_2(\cdot))$ and $\Pi_2' := \Pi_5(k_0 \oplus \Pi_4(\cdot))$, and $F(y) := \Pi_1'(y) \oplus y$
5: Find a t-collision of F and let v be its value
6: **amplify with** $2^{n/2}/\sqrt{\log n}$ **iterations**
7: Pick a random query $(x, E(x))$ in the database
▷ This step requires QRACM access to the stored known-plaintext queries
8: Compute $x' := \Pi_1(k_0 \oplus x)$ and $E'(x') := \Pi_6^{-1}(k_0 \oplus E(x))$
9: Compute $k_1 := E'(x') \oplus \Pi_2'(x' \oplus v)$
10: Check if k_0, k_1 is the correct key using two known plaintext-ciphertext pairs
11: **Good outcome:** k_0, k_1 is the correct key
12: **end amplify**
13: **Good outcome:** k_0, k_1 is the correct key
14: **end amplify**

The idea of Algorithm 5 is that, provided that k_0 has been correctly guessed, the inner quantum search only needs to examine $2^n/\log n$ input-output pairs for the inner cipher $E'(x) := k_1 \oplus \Pi_2'(k_1 \oplus \Pi_1'(k_1 \oplus x))$, in order to identify the correct key k_1. We can provide these input-output pairs from $2^n/\log n$ classical known-plaintext queries, which remain fixed throughout the algorithm, and which we process during the quantum search into pairs for E' (Step 8).

The attack will now require $\frac{2^n}{\log n}$ classical known-plaintext queries, $\frac{2^n}{\log n}$ QRACM to hold these queries (we only need fast quantum access), $\frac{2^n}{\sqrt{\log n}}$ quantum time and $\frac{2^{n/2}}{\sqrt{\log n}}$ QRAQM to compute the multicollision at Step 5 using the algorithm of [24].

6 Classification of Iterated Even-Mansour Schemes with Two Keys

In this section, we give the best time-data trade-offs for quantum attacks on iterated Even-Mansour schemes with two keys and 3 to 4 rounds, depending on the sequence of keys. These attacks use classical queries, either chosen-plaintext or known-plaintext.

For these trade-offs, we only need to consider the equivalence classes from [17], where two key schedules can be equivalent by swapping the roles of k_0 and k_1, or by reversing the order of the keys (which swaps the roles of encryption and decryption). Our results are summarized in Table 4. We observe that the key schedule k_0, k_1, k_0, k_1, k_0, which is the most natural one, is the *hardest* case regarding known attacks, and the only one for which no attack of time complexity (exponentially) less than N was known prior to our work.

Table 4. Classification of quantum attacks on iterated Even-Mansour with two keys. Time complexity is given in the number of queries to public permutations. Quantum memory includes QRAQM and QRACM (more details are given in Table 2).

Key schedule	Classical data	Quantum time	Memory	Reference
k_0, k_1, k_1, k_1	negl.	$\sqrt{N}$	negl.	Grover's search
k_0, k_1, k_0, k_0	negl.	$\sqrt{N}$	negl.	Grover's search
k_0, k_1, k_1, k_0	negl.	$N^{2/3}$	$N^{2/3}$	Meet-in-the-middle
k_0, k_1, k_0, k_1	$D \leq N^{2/3}$	$N \log N/\sqrt{D}$	negl.	Offline-Simon [9]
k_0, k_1, k_1, k_1, k_1	negl.	$\sqrt{N}$	negl.	Grover's search
k_0, k_1, k_0, k_0, k_0	negl.	$\sqrt{N}$	negl.	Grover's search
k_0, k_0, k_1, k_0, k_0	negl.	$\sqrt{N}$	negl.	Grover's search
k_0, k_0, k_1, k_1, k_1	negl.	$N^{2/3}$	$N^{2/3}$	Meet-in-the-middle
k_0, k_1, k_1, k_0, k_0	negl.	$N^{2/3}$	$N^{2/3}$	Meet-in-the-middle
k_0, k_1, k_1, k_1, k_0	$D \leq N^{2/3}$	$N \log N/\sqrt{D}$	negl.	Offline-Simon [8]
k_0, k_1, k_0, k_1, k_0	$D \leq N^{1/3}$	$N/\sqrt{D}$	D	Section 3.3
	$N^{1/3} \leq D \leq N^{2/3}$	$N^{5/6}$	$N^{1/3}$	Section 3.3
	$N^{2/3} \leq D \leq N^{7/9}$	$N^{7/6}/\sqrt{D}$	D	Section 4.2
k_0, k_1, k_0, k_1, k_1	$D \leq N^{2/3}$	$N \log N/\sqrt{D}$	negl.	Offline-Simon [9]
k_0, k_1, k_1, k_0, k_1	$D \leq N^{2/3}$	$N \log N/\sqrt{D}$	negl.	Offline-Simon [9]

In the following, we detail the three families of attacks based on previous works which appear in the table: the offline-Simon attacks, the attacks based on Grover's search, and the Meet-in-the-middle attacks.

Offline-Simon. Consider the key sequence k_0, k_1, k_0, k_1 on 3 rounds, which is represented in Figure 8.

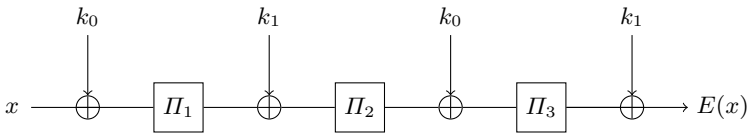

Fig. 8. Two-key iterated Even-Mansour cipher with 3 rounds.

By defining $E^1_{k_1}(x) := \Pi_2(k_1 \oplus \Pi_1(x))$ and $E^2_{k_1}(x) := k_1 \oplus \Pi_3(x)$, we can rewrite the cipher as Figure 9.

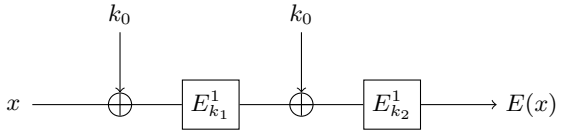

Fig. 9. Extended FX construction from a 2-key 3-EM construction.

This construction is named *Extended FX* in [9], and it is a construction on which the offline-Simon algorithm is applicable. By Theorem 3 in [9], this key-recovery attack uses D classical chosen-plaintext queries, only polynomial quantum memory and has a quantum time complexity $\mathcal{O}\Big(\log N(D + N/\sqrt{D})\Big)$ for any $D \leq N$. The only disadvantage of the offline-Simon attack compared to other attacks studied in this paper is that it requires classical chosen-plaintext queries, instead of known-plaintext queries.

When the cipher can be rewritten as an FX construction, i.e., removing the second block cipher call $E^2_{k_1}(x)$ from Figure 9, then the original offline-Simon attack from [8] applies and reaches the same time-data trade-off.

Grover's Search. In several cases, guessing one of the two keys reduces the cipher to a single key addition. Therefore, we can break it using two known plaintext-ciphertext pairs, using a Grover's search on the guessed key.

Meet-in-the-middle. In the remaining cases, we reduce the problem to a collision search problem, which we solve using a quantum walk [2]. The data complexity is also negligible; in all cases we use two known plaintext-ciphertext pairs $(x, E(x)), (x', E(x'))$.

For the key-schedule k_0, k_1, k_1, k_0, we take the list of all $(\alpha, \Pi(\alpha))$ for $\alpha \in \{0,1\}^n$ and the list of all $\Pi_1(x \oplus k_0), \Pi_3^{-1}(k_0 \oplus E(x))$ for $k_0 \in \{0,1\}^n$, and we merge them using the condition:

$$\Pi_1(x \oplus k_0) \oplus \alpha = \Pi_3^{-1}(k_0 \oplus E(x)) \oplus \Pi(\alpha) \ . \tag{20}$$

Here the quantity $\Pi_1(x \oplus k_0) \oplus \alpha$ will correspond to k_1. In the quantum setting, similar to our quantum multibridge algorithm, we use a quantum walk on a

product Johnson graph, where each vertex is a pair of subsets for α and k_0. The marked condition is that one of the pairs satisfying the condition leads to the correct key (k_0, k_1) (checked using the second plaintext-ciphertext pair). As this is a claw-finding problem for (α, k_1) with a single solution, a similar complexity analysis leads to a quantum time complexity $N^{2/3}$ and memory $N^{2/3}$ (QRAQM). There is also a time-memory trade-off, but since we are focused on time-data trade-offs instead, we do not include it in Table 4.

For the key-schedule k_0, k_0, k_1, k_1, k_1, the attack is easier, as it is exactly a claw-finding problem with a single solution, solved by Ambainis' quantum walk algorithm [2] in time $N^{2/3}$ and memory $N^{2/3}$ (QRAQM).

For the key-schedule k_0, k_1, k_1, k_0, k_0, we use the same attack as k_0, k_1, k_1, k_0 except that there is an additional round to decrypt using k_0. The complexity is the same.

7 Conclusion

In this paper, we presented new quantum attacks on iterated Even-Mansour ciphers with two keys. Our new 4-round attacks improve asymptotically by an exponential factor ($2^{\alpha n}$) on exhaustive key search, while our new 6-round attack improves by a logarithmic factor $\sqrt{\log n}$, i.e., constant in practice. While the former is permitted by quantum collision search and quantum walks, the latter comes from solving a multicollision problem.

Interestingly, these new attacks can rely on classical *known-plaintext* queries only, i.e., data limitations very similar to the classical setting. This is not completely unsurprising. Indeed, with a total key material of $2n$ bits, the cost of exhaustive search stands at 2^n, while the amount of classical data is 2^n. This means that one can still fit a large classical data complexity in the quantum attack. As an example, our best complexity for 2-key 4-EM ($2^{7n/9}$) is obtained with $2^{7n/9}$ classical data. Designing better attacks using Q2 queries, while less relevant for practical security, would be interesting to compare to quantum proofs of security in the Q2 setting [5].

Another important question is whether the memory footprint of these attacks can be improved. While our attacks are able to switch between different memory models, a common point is that they require exponential-size quantum memory, whether QRACM, QRAQM, or a large number of qubits. As an example, attacking 4-key 2-EM with classical memory only (like it is possible for the single-round EM) would be very interesting.

Finally, the 4-key 2-EM structure, where the two keys alternate, appears to be the only one for which the best quantum time complexity is only $2^{7n/9}$: all other key schedules have an attack in time $2^{2n/3}$ at most. This situation could motivate further work into the tightness of our quantum multibridge attack.

Acknowledgements. This work has been partially supported by the French Agence Nationale de la Recherche through the France 2030 program under grant agreement No. ANR-22-PETQ-0008 PQ-TLS and through the QATS project under

Contract ANR-24-CE39-7894-01, and by the Agence de l'innovation de défense (AID) through the ASCON-CAT project.

References

1. Alagic, G., Bai, C., Katz, J., Majenz, C.: Post-quantum security of the Even-Mansour cipher. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 458–487. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_17
2. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007). https://doi.org/10.1137/S0097539705447311, https://doi.org/10.1137/S0097539705447311
3. Anand, R., Ghosh, S., Isobe, T., Shiba, R.: Quantum key recovery attacks on 4-round iterated Even-Mansour with two keys. In: Mouha, N., Nikiforakis, N. (eds.) ISC 2024, Part I. LNCS, vol. 15257, pp. 87–103. Springer, Cham (Oct 2024). https://doi.org/10.1007/978-3-031-75757-0_5
4. Avanzi, R., Banik, S., Dunkelman, O., Eichlseder, M., Ghosh, S., Nageler, M., Regazzoni, F.: The QARMAv2 family of tweakable block ciphers. IACR Trans. Symm. Cryptol. **2023**(3), 25–73 (2023). https://doi.org/10.46586/tosc.v2023.i3.25-73
5. Bai, C., Esmaili, M., Mantri, A.: Quantum security analysis of the key-alternating ciphers. Cryptology ePrint Archive, Report 2025/945 (2025), https://eprint.iacr.org/2025/945
6. Bogdanov, A., Knudsen, L.R., Leander, G., Standaert, F.X., Steinberger, J.P., Tischhauser, E.: Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations - (extended abstract). In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 45–62. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_5
7. Bonnetain, X., Chailloux, A., Schrottenloher, A., Shen, Y.: Finding many collisions via reusable quantum walks: Application to lattice sieving. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 221–251. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_8
8. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: The offline Simon's algorithm. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 552–583. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-34578-5_20
9. Bonnetain, X., Schrottenloher, A., Sibleyras, F.: Beyond quadratic speedups in quantum attacks on symmetric schemes. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 315–344. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_12
10. Bozilov, D., Eichlseder, M., Knezevic, M., Lambin, B., Leander, G., Moos, T., Nikov, V., Rasoolzadeh, S., Todo, Y., Wiemer, F.: PRINCEv2 - more security for (almost) no overhead. In: Dunkelman, O., Jacobson, Jr., M.J., O'Flynn, C. (eds.) SAC 2020. LNCS, vol. 12804, pp. 483–511. Springer, Cham (Oct 2020). https://doi.org/10.1007/978-3-030-81652-0_19
11. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002). https://doi.org/10.1090/conm/305/05215

12. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: Lucchesi, C.L., Moura, A.V. (eds.) LATIN 1998. LNCS, vol. 1380, pp. 163–169. Springer, Berlin, Heidelberg (Apr 1998). https://doi.org/10.1007/bfb0054319
13. Chailloux, A., Naya-Plasencia, M., Schrottenloher, A.: An efficient quantum collision search algorithm and implications on symmetric cryptography. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 211–240. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_8
14. Chen, S., Steinberger, J.P.: Tight security bounds for key-alternating ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer, Berlin, Heidelberg (May 2014). https://doi.org/10.1007/978-3-642-55220-5_19
15. Daemen, J.: Limitations of the Even-Mansour construction (rump session). In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT'91. LNCS, vol. 739, pp. 495–498. Springer, Berlin, Heidelberg (Nov 1993). https://doi.org/10.1007/3-540-57332-1_46
16. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Key recovery attacks on 3-round Even-Mansour, 8-step LED-128, and full AES2. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 337–356. Springer, Berlin, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42033-7_18
17. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Cryptanalysis of iterated Even-Mansour schemes with two keys. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 439–457. Springer, Berlin, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45611-8_23
18. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in cryptography: The Even-Mansour scheme revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 336–354. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_21
19. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. Journal of Cryptology **10**(3), 151–162 (Jun 1997). https://doi.org/10.1007/s001459900025
20. Flajolet, P., Odlyzko, A.M.: Random mapping statistics. In: Quisquater, J.J., Vandewalle, J. (eds.) EUROCRYPT'89. LNCS, vol. 434, pp. 329–354. Springer, Berlin, Heidelberg (Apr 1990). https://doi.org/10.1007/3-540-46885-4_34
21. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC. pp. 212–219. ACM Press (May 1996). https://doi.org/10.1145/237814.237866
22. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Berlin, Heidelberg (Sep / Oct 2011). https://doi.org/10.1007/978-3-642-23951-9_22
23. Hosoyamada, A., Sasaki, Y.: Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In: Smart, N.P. (ed.) CT-RSA 2018. LNCS, vol. 10808, pp. 198–218. Springer, Cham (Apr 2018). https://doi.org/10.1007/978-3-319-76953-0_11
24. Hosoyamada, A., Sasaki, Y., Tani, S., Xagawa, K.: Improved quantum multicollision-finding algorithm. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 350–367. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_19

25. Isobe, T., Shibutani, K.: New key recovery attacks on minimal two-round Even-Mansour ciphers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 244–263. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_9
26. Jeffery, S.: Frameworks for Quantum Algorithms. Ph.D. thesis, University of Waterloo, Ontario, Canada (2014), http://hdl.handle.net/10012/8710
27. Kachigar, G., Tillich, J.P.: Quantum information set decoding algorithms. In: Lange, T., Takagi, T. (eds.) Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017. pp. 69–89. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59879-6_5
28. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 207–237. Springer, Berlin, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53008-5_8
29. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: ISIT. pp. 2682–2685. IEEE (2010). https://doi.org/10.1109/ISIT.2010.5513654
30. Kuwakado, H., Morii, M.: Security on the quantum-type even-mansour cipher. In: ISITA. pp. 312–316. IEEE (2012)
31. Leander, G., May, A.: Grover meets Simon - quantumly attacking the FX-construction. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 161–178. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_6
32. Leurent, G., Sibleyras, F.: Low-memory attacks against two-round Even-Mansour using the 3-XOR problem. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 210–235. Springer, Cham (Aug 2019). https://doi.org/10.1007/978-3-030-26951-7_8
33. Liu, Q., Zhandry, M.: On finding quantum multi-collisions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 189–218. Springer, Cham (May 2019). https://doi.org/10.1007/978-3-030-17659-4_7
34. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. SIAM J. Comput. **40**(1), 142–164 (2011). https://doi.org/10.1137/090745854, https://doi.org/10.1137/090745854
35. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
36. Santoli, T., Schaffner, C.: Using simon's algorithm to attack symmetric-key cryptographic primitives. Quantum Inf. Comput. **17**(1&2), 65–78 (2017). https://doi.org/10.26421/QIC17.1-2-4, https://doi.org/10.26421/QIC17.1-2-4
37. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th FOCS. pp. 124–134. IEEE Computer Society Press (Nov 1994). https://doi.org/10.1109/SFCS.1994.365700
38. Simon, D.R.: On the power of quantum computation. In: 35th FOCS. pp. 116–123. IEEE Computer Society Press (Nov 1994). https://doi.org/10.1109/SFCS.1994.365701
39. Tani, S.: Claw finding algorithms using quantum walk. Theoretical Computer Science **410**(50), 5285–5297 (2009)