An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography

André Chailloux, María Naya-Plasencia, and André Schrottenloher

Inria Paris, France
firstname.lastname@inria.fr

Abstract. The cryptographic community has widely acknowledged that the emergence of large quantum computers will pose a threat to most current public-key cryptography. Primitives that rely on order-finding problems, such as factoring and computing Discrete Logarithms, can be broken by Shor's algorithm ([50]).

Symmetric primitives, at first sight, seem less impacted by the arrival of quantum computers: Grover's algorithm [32] for searching in an unstructured database finds a marked element among 2^n in time $\widetilde{O}(2^{n/2})$, providing a quadratic speedup compared to the classical exhaustive search, essentially optimal. Cryptographers then commonly consider that doubling the length of the keys used will be enough to maintain the same level of security.

From similar techniques, quantum collision search is known to attain $\widetilde{O}(2^{n/3})$ *query complexity* [21], compared to the classical $O(2^{n/2})$. However, as Bernstein pointed out in [9], this quantum speedup is illusory: the actual quantum computation performed is actually more expensive than in the classical algorithm.

In this paper, we investigate quantum collision and multi-target preimage search and present a new algorithm, that uses the amplitude amplification technique. As such, it relies on the same principle as Grover's search. Our algorithm is the first to propose a time complexity that improves upon $O(2^{n/2})$, in a simple setting with a single processor. This time complexity is $\widetilde{O}(2^{2n/5})$ (equal to its query complexity), with a polynomial quantum memory needed ($O(n)$), and a small classical memory complexity of $\widetilde{O}(2^{n/5})$. For multi-target preimage attacks, these complexities become $\widetilde{O}(2^{3n/7})$, $O(n)$ and $\widetilde{O}(2^{n/7})$ respectively. To the best of our knowledge, this is the first proof of an actual quantum time speedup for collision search. We also propose a parallelization of these algorithms.

This result has an impact on several symmetric cryptography scenarios: we detail how to improve upon previous attacks for hash function collisions and multi-target preimages, how to perform an improved key recovery in the multi-user setting, how to improve the collision attacks on operation modes, and point out that these improved algorithms can serve as basic tools for some families of cryptanalytic techniques.

In the end, we discuss the implications of these new attacks on post-quantum security.

Keywords: post-quantum cryptography, symmetric cryptography, collision search, amplitude amplification.

1 Introduction

The emergence of large-scale quantum computing devices would have enormous consequences in physics, mathematics and computer science.

While *quantum hegemony* has yet to be achieved by these machines, the field of post-quantum cryptography has become very active in the last twenty years, as it is of foremost importance to achieve *today* security against possible adversaries from tomorrow. As a consequence, post-quantum asymmetric primitives are being developed and standardized, to protect public-key cryptography against the ravages of Shor's period-finding algorithm ([50]), that provides an exponential advantage to a quantum adversary compared to all known classical factorization algorithms.

Symmetric Cryptography in the Quantum World. In the symmetric setting, Grover's algorithm can speed up quadratically the classical exhaustive key search. As a result, ideal ciphers with k-bit keys would provide only $k/2$-bit security in a post-quantum world. The confidence we have on real symmetric primitives is based on cryptanalysis, i.e. the more we analyze a primitive without finding any weakness, the more trust we can put in it. Until recently, little was known on how quantum adversaries could try to attack symmetric primitives. Therefore, as little was known about their security and confidence they should inspire in a quantum world.

This is why turning classical attacks into quantum attacks was studied in [37] and [35]. By transposing the weaknesses of an encryption function to the post-quantum world, it is indeed possible to improve on the naive, all-purpose Grover search. How classical attacks can be "quantized" requires, however, a careful analysis.

Besides, if the adversary has stronger capacities than the mere access to a quantum computing device (e.g, if she can ask superposition chosen-plaintext queries), an exponential speedup has been shown to occur for some constructions. This was first noted by Kuwanado and Morii against the Even-Mansour construction ([43]) and the three-round Feistel ([42]), later extended to slide attacks and modes of operation for MACs ([36]). All these attacks use Simon's algorithm [51].

Quantum Collision and Multi-target Preimage Search. In a classical setting, it is well known that finding a collision for a random function H on n bits, i.e. a pair x, y with $x \neq y$ such that $H(x) = H(y)$, costs $O\left(2^{n/2}\right)$ in time and queries [49]. A parallelization of this algorithm was proposed in [48], that has a product of time and memory complexities of also $O\left(2^{n/2}\right)$.

In a quantum setting, an algorithm was presented by Brassard, Høyer and Tapp in [21] that, given superposition query access to a (2-to-1) function H

on n bits, outputs a collision using $\widetilde{O}\left(2^{n/3}\right)$ superposition queries to H. This algorithm is optimal, but only in terms of *query complexity*, while its product of time and memory complexities is as high as $\widetilde{O}\left(2^{2n/3}\right)$ and makes it non-competitive when compared to the classical attack. This was pointed out by Bernstein in 2009 ([9]), who claimed that quantum computers would not make any improvement over on-purpose hardware for collision search.

Regarding the multi-target preimage search, i.e. given t values of $H(x_i)$ for i from 1 to t, find one out of the t values of the x_i, the best classical algorithm finds a preimage in time $O\left(2^{n-t}\right)$. In the quantum setting, the best algorithm takes time $\widetilde{O}(2^{n/2})$, it consists in fact of finding the preimage of a single *chosen* target with Grover's algorithm.

1.1 Contributions.

The contributions we present in this paper are two folded:

Improved Algorithms for Collision and Multi-target Preimage Search. First, we propose a new quantum algorithm for collision search, based on amplitude amplification, which runs in real time $\widetilde{O}\left(2^{2n/5}\right)$ with a single quantum processor, uses $O(n)$ qubits of memory, and $\widetilde{O}\left(2^{n/5}\right)$ bits of classical storage, accessed via a classical processor. The algorithm can be adapted to solve the multi-target preimage problem, with a running time $\widetilde{O}\left(2^{3n/7}\right)$, the same quantum requirements and $\widetilde{O}\left(2^{n/7}\right)$ bits of classical storage.

We also extend these results if quantum parallelization is allowed. These quantum algorithms are the first ones to significantly improve on the best classical algorithms for those two problems. These results also solve an open problem and contradict a conjecture on the complexity of quantum collision and multi-target preimage search, as we will detail in section 7.2.

Implications of these Algorithms. We have studied the impact of these new algorithms on several cryptographic settings, and obtained the following conclusions:

- Hash functions: We are able to improve the best known collision and multi-target preimage attacks when the attacker has only access to a quantum computer.
- Multi-user setting: We are able to improve the best known attacks in a multi-user setting, i.e. recover one key out of many, thanks to the multi-target preimage search improved algorithm. The model for the attacker here is also not very strong, and we only suppose she has access to a quantum computer.
- Operation Modes: Considering collision attacks on operation modes, we are able to improve them with our new algorithms. In this case, the attacker is placed in a more powerful model: she can make superposition queries to a quantum cryptographic oracle. The question of a new data limit enforcement is raised.

- Bricks for cryptanalysis techniques: we show how these algorithms can be used as building blocks in more complex cryptanalytic techniques.

We also discuss the implications of these attacks with respect to security bounds for symmetric cryptographic primitives.

Organisation. In the next section, we detail some security notions that will be considered in our applications and some basic notions of quantum computing. In Section 3, we present the considered problems: collision and multi-target preimage search, and we report the state-of-the-art of the previous best known quantum and classical algorithms for solving them. We also present some cryptographic scenarios where these problems are shown to be useful. In Section 4, we develop a prerequisite building block of our algorithms, while Section 5 is dedicated to detail the new algorithms and possible trade-offs. In Section 6 we analyze the impact of these algorithms with respect to the cryptographic scenarios previously presented. A discussion on our results and a conclusion are provided in Section 7. In the auxiliary supporting material appended to this submission, we deal with the algorithmic imperfections of the amplitude amplification algorithm.

2 Preliminaries

This section describes some concepts that will be needed for presenting our results: we first provide some classical security notions. Next we describe the two models most commonly considered for quantum adversaries, as both will be considered in applications (Section 6). Finally, we will briefly describe the basic quantum computing notions that we will need in order to explain our new algorithms in section 5.

2.1 Some Classical Security Notions

In this section we briefly describe some notions from symmetric key cryptography that will be used through the paper.

Key Recovery Attack. Consider a cipher E_K, that is a pseudo-random permutation parameterized by a secret key K of size k. This cipher takes as input a plaintext of size n and generates a ciphertext of the same size. In the common known-plaintext setting (KPA), the attacker gets access to some pairs of plaintexts and ciphertexts (P_i, C_i). Sometimes the attacker is also allowed to choose the plaintexts: this is called chosen-plaintext attacks (CPA).

It is always possible to perform an exhaustive search on the key and to find the correct one as the one that verifies $C_i = E_K(P_i)$ for all i. The cost of this is 2^k encryptions, and this is the security an ideal cipher provides: the best attack is the generic attack. Therefore, a cipher is *broken* if we are able to recover its key with a complexity smaller than 2^k. The data complexity will be the number of

calls to the cryptographic oracle E_K, i.e. the number of pairs (P_i, C_i) needed to successfully perform the attack; the time complexity is the overall time needed to recover the key, and the memory complexity is the amount of memory needed to perform the attack.

Distinguisher and Plaintext Recovery. Key recovery attacks are the strongest, but being able to distinguish the generated ciphertexts from random values is also considered as a weakness. Moreover, when the attacker only captures ciphertexts, she shouldn't able to recover information of any kind on the corresponding plaintexts.

Modes of Operations. In order to be able to treat messages of different lengths and to provide specific security properties, as confidentiality and integrity, block ciphers are typically used in *operation modes*. One of the security properties that these modes should offer is, for instance, not to allow an attacker to identify when the same two blocks have been encrypted under the same key, without having to change the key for each block (which wouldn't be very efficient). Some popular modes are Cipher Block Chaining, CBC [27], or Counter Mode, CTR [26]. It is also possible to build authenticated encryption primitives by using authentication modes, as the Offset Codebook Mode, OCB [40] proposed by Krovetz and Rogaway. Their securities have been widely studied in the classical setting ([8]), as well as recently in a post-quantum setting ([5]).

A plaintext m is split in blocks $m_0 \dots m_{l-1}$, that will be encrypted with the help of the cipher E_K and combined; the ciphertext is $c = c_0 \dots c_{l-1}$.

CBC. The Code Block Chaining (CBC) mode of operation defines the ciphertext blocks as follows: $c_0 = E_K(m_0 \oplus IV)$ and for all $i \leq l-1$:

$$c_i = E_K(m_i \oplus c_{i-1})$$

where IV is a public initial value.

The block size being n, some restrictions on the maximal number of blocks encrypted under the same key must be enforced. Indeed, the birthday paradox implies that after recovering $2^{n/2}$ encrypted blocks, there is a high (and constant) probability that two of them are equal, leading to:

$$E_K(m_i \oplus c_{i-1}) = E_K(m_j \oplus c_{j-1}) \ .$$

And since E_K is a permutation, we get $m_i \oplus c_{i-1} = m_j \oplus c_{j-1}$ hence $m_i \oplus m_j$, the XOR of two plaintext blocks, from the knowledge of the ciphertexts.

CTR. In the counter mode (CTR), blocks m_i are encrypted as $c_i = E_K(IV \oplus i) \oplus m_i$ where IV is an initial public value, and i is a counter initialized to zero. As all the inputs of the encryption function are different, we won't have collisions due to the birthday paradox as in the CBC case, but this lack of collisions can exploited to distinguish the construction if more than the $2^{n/2}$ recommended bound of data was generated with the same key.

2.2 Quantum Adversary Models

In this section we describe and justify the two models most commonly considered for quantum adversaries. The application scenarios described in section 6 will use both of them.

Model Q_1. The adversary has access to a quantum computer: this is the case, for instance, in [15, 19, 56, 52]. The adversary can query a quantum random oracle with arbitrary superpositions of the inputs, but is only able to make classical queries to a classical encryption oracle (and therefore no quantum superposition queries to the cryptographic oracle).

Model Q_2. In this case, the adversary is allowed to perform quantum superposition queries to a remote quantum cryptographic oracle (qCPA): she obtains a superposition of the outputs. This model has been considered for instance in [23, 17, 54, 36, 30, 16]. This is a strong model, but it has the advantages of being simple, inclusive of any possible intermediate and more realistic model, and achievable. In particular, in several of these publications, secure constructions were provided for this scenario.

2.3 Quantum Computing

In this section we provide some basic notions from quantum computing that will be used through the paper. The interested reader can see [47] for a detailed introduction to quantum computing.

Quantum Oracles for Functions. Any function $f : \{0,1\}^n \to \{0,1\}^m$ with a known circuit description can be efficiently implemented as a quantum unitary O_f on $n+m$ qubits, with:

$$O_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle \quad .$$

The quantum running time of O_f is twice [1] the running time of f.

Projection Oracle. Let P a projector acting on n qubits. We define O_P as the following unitary acting on $n+1$ qubits

$$O_P(|\psi\rangle |b\rangle) := \begin{cases} |\psi\rangle |b \oplus 1\rangle & \text{if } |\psi\rangle \in Im(P) \\ |\psi\rangle |b\rangle & \text{if } |\psi\rangle \in Ker(P) \end{cases} .$$

The above expression defines O_P on a basis of the $n+1$ qubit pure states and O_P is therefore defined for all states by linearity.

[1] Computing f makes use of ancillary (additional) qubits. Properly initialized to $|0\rangle$, those end up in a state $|g(x)\rangle$ that cannot be simply dismissed: instead, by *uncomputing*, we can restore these qubits to their initial state $|0\rangle$ and make sure that the oracle has no side-effects.

Amplitude Amplification. One of the main tools we will use in our algorithms is the quantum amplitude amplification routine.

Theorem 1 ([20], Quantum amplitude amplification). *Let P a projector acting on n qubits and O_P a projection oracle for P. Let $\mathcal{A}$ be a quantum unitary that produces a state $|\phi\rangle = \alpha |\phi_P\rangle + \beta \left|\phi_P^{\perp}\right\rangle$ where $|\phi_P\rangle \in Im(P)$ and $\left|\phi_P^{\perp}\right\rangle \in Ker(P)$. Notice that $tr(P|\phi\rangle\langle\phi|) = |\alpha|^2$. We note $|\alpha| = \sin(\theta)$ for some $\theta \in [0, \pi/2]$. There exists a quantum algorithm that:*

- *Consists of exclusively $N = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rfloor$ calls to $O_P, O_P^{\dagger}, \mathcal{A}, \mathcal{A}^{\dagger}$ and a final measurement.*
- *Produces a quantum state close to $|\phi_P\rangle$. For simplicity, we do not specify the error here and dedicate instead the supplementary material to it.*

The algorithm $\mathcal{A}$ is called the setup and the projection P the projector of the quantum amplification algorithm. This whole procedure will be denoted

$$\texttt{QAA}(\texttt{setup}, \texttt{proj}) = \texttt{QAA}(\mathcal{A}, P)$$

and its running time is

$$N\left(|\mathcal{A}|_{RT} + |O_P|_{RT}\right) \quad .$$

where the notation $|\cdot|_{RT}$ represents the running time of the respective algorithms. If both $\mathcal{A}$ and O_P can be done in time polynomial in n, the above is $\widetilde{O}(N)$.

This projection P can be sometimes characterized by a test function f, such that $|x\rangle \in Im(P)$ when $f(x) = 1$ and $|x\rangle \in Ker(P)$ when $f(x) = 0$. Amplitude amplification can be seen as a generalization of Grover's algorithm. Let us briefly show how to retrieve it.

Grover's Algorithm. We are given an efficiently computable function $f : \{0,1\}^n \mapsto \{0,1\}$ and we want to find an element x such that $f(x) = 1$. We take P such that $|x\rangle \in Im(P)$ when $f(x) = 1$ and $|x\rangle \in Ker(P)$ when $f(x) = 0$. O_P can be constructed with a single call to O_f. We use as setup the algorithm $\mathcal{A}$ that produces the state $|\phi\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle$. In order to produce $|\phi\rangle$, we perform a Hadamard operation on each qubit, which is very efficient.

We write $|\phi\rangle = \frac{1}{2^{n/2}} \sum_{x:f(x)=1} |x\rangle + \frac{1}{2^{n/2}} \sum_{x:f(x)=0} |x\rangle$. We have $tr(P|\phi\rangle\langle\phi|) = \frac{|\{x:f(x)=1\}|}{2^n}$. Using the above quantum amplitude amplification procedure $\texttt{QAA}(\mathcal{A}, P)$, and by measuring the obtained state, we can find with high probability an element x such that $f(x) = 1$ in time $\widetilde{O}\left(\sqrt{\frac{2^n}{|\{x:f(x)=1\}|}}\right)$.

For most applications, e.g quantum exhaustive key search, there is only one "marked" element x such that $f(x) = 1$ (e.g, the key). Then Grover search attains a complexity $\widetilde{O}\left(\sqrt{2^n}\right)$.

Quantum Query, Memory and Time Complexity. Most of the complexity lower bounds on quantum algorithms in the literature, as well as the algorithms that meet these bounds, are based on *query complexity*. As such, they count the number of oracle queries O_f used, where O_f is a quantum oracle for a function f or more generally the data that is being accessed.

Notice that classical queries are a particular case of superposition queries, so we consider them alike in what follows.

However, *query complexity* can be completely different from *time complexity*: the latter represents the number of elementary quantum gates (unitaries) successively applied to a qubit or a qubit register. It has the same flavor as classical time complexity, since it counts elementary operations applied sequentially.

We emphasize that *memory complexity* has a different meaning in the quantum and the classical setting. While classical memory is thought of as a database with fast access, *quantum memory* denotes the number of qubits in the circuit. Having more qubits means that more operations can be applied in parallel, hence decreases the time complexity: it rather corresponds to classical *parallelization*.

3 State-of-the-Art on Collision and Multi-target Preimage Search

The two problems that we consider in a quantum setting, collision search and multi-target preimage search, are described in this section. We also briefly describe the best known classical algorithms for solving them and their complexities, as well as the previously best known quantum algorithms, that we will improve in section 5. We will provide a discussion on the comparison of both previous algorithms. In the end of this section we additionally provide some examples of common applications of this problems on cryptanalysis.

3.1 Studied Problems

In this work we consider the two following problems:

Problem 1 (Collision finding). Given access to a random function $H : \{0,1\}^n \to \{0,1\}^n$, find $x, y \in \{0,1\}^n$ with $x \neq y$ such that $H(x) = H(y)$.

We consider here a random function which models best the cryptographic functions (encryption functions or hash functions) that we want to study.

Problem 2 (Multi-target preimage search). Given access to a random permutation $H : \{0,1\}^n \to \{0,1\}^n$ and a set $T = \{y_1, \ldots, y_{2^t}\}$, find the preimage of one of the y_i by E *i.e.* find $i \in \{1, \ldots, \ell\}$ and $x \in \{0,1\}^n$ such that $E(x) = y_i$.

The above problem can also be considered when replacing H with a random function.

Previous quantum algorithms in the literature ([20]) for Prob. 1 and 2 consider sometimes the case of r-to-1 functions. Although we restrict ourselves to the

case of random functions and permutations, which is relevant in cryptographic applications, we remark that the algorithms presented below could be rewritten for r-to-1 functions.

3.2 Classical Algorithms to Solve Them

Collision search. The birthday paradox states that if we draw at random $2^{n/2}$ elements $x_i \in \{0,1\}^n$ we will find a collision between two of their images, i.e. $H(x_i) = H(x_j)$, with good probability (i.e. 0.5), and a collision can be found with $O(2^{n/2})$ time and memory. Pollard's rho algorithm [49] allows to reduce the memory complexity to a negligible amount while keeping the same time complexity. No classical algorithm with a single processor exists for finding collisions on a set of 2^n elements with a lower time complexity than $O(2^{n/2})$.

Parallelizing collision search. In [48] a method for reducing the time complexity efficiently through parallelization is proposed. The total amount of computations is slightly increased, and the time-space product is not smaller than $O(2^{n/2})$, but the speed up will be linear. The method is based in considering a common list were all found distinguished points will be stored, until a collision on them is found. The time complexity becomes $O(2^{n/2}/m + 2.5\theta)$, for a case where all collisions are useful and must be located when considering m processors, and θ is the proportion of distinguished points, that will have a direct impact in the memory needs.

Multi-target preimage attacks. With respect to this second problem, the best classical algorithm finds one out of $\ell = 2^t$ targets with an exhaustive search in $\Omega(2^{(n-t)})$ (see for instance [6]). This complexity can be trivially derived by the fact that the probability of finding one out of the 2^t preimages is $\frac{2^t}{2^n}$.

3.3 Previous Quantum Algorithms

Quantum Algorithms for the Collision Problem The quantum collision search problem was first studied by Brassard, Høyer and Tapp ([21]). Using Grover's algorithm as a subroutine, they showed that the collision problem for a 2-to-1 function f could be solved using $\widetilde{O}(2^{n/3})$ queries to O_f and $\widetilde{O}(2^{n/3})$ *quantum memory*. After, there has been many results on query lower bounds for the collision problem, ([1, 2, 41]), until a bound $\Omega(2^{n/3})$ was reached. Zhandry also extended the collision problem to random functions, which is relevant in a cryptographic setting ([55]), and proved that this bound still held.

Another related and well studied problem is *element distinctness*, where the question is to decide whether the outputs of the function f are all distinct (or, equivalently, to find a collision if there is at most one). In particular, Ambainis ([3]) presented a quantum walk algorithm for this problem and showed a *time complexity* of $\widetilde{O}(2^{2n/3})$, using $\widetilde{O}(2^{2n/3})$ quantum bits of memory. In [1] $\Omega(2^{2n/3})$ was shown to be a query lower bound for this problem, so those results are

essentially optimal. It is known that element distinctness can be reduced to collision by gaining a root in the time complexity, which gives an essentially optimal quantum time and memory of $\widetilde{O}(2^{n/3})$.

Here, we show the original algorithm for collision search from ([21]), that uses Grover. This algorithm has query complexity $\widetilde{O}(2^{n/3})$ but running time $\widetilde{O}(2^{2n/3})$. It is also possible to reduce the running time of the algorithm below to $\widetilde{O}(2^{n/3})$ by using $\widetilde{O}(2^{n/3})$ quantum processors in parallel.

Algorithm 1: Collision search in a 2-to-1 function using Grover ([21])

Input. Quantum query access to the 2-to-1 function $H : \{0,1\}^n \to \{0,1\}^n$ (oracle O_H).
Membership oracle. We query H on an arbitrary set of $2^{n/3}$ values T, obtaining a set $H(T)$. The algorithm queries a quantum oracle O for testing if $H(x) \in H(T)$. Either each query to O needs quantum time $\widetilde{O}(|T|) = \widetilde{O}(2^{n/3})$, either each is performed in $O(1)$ but the implementation requires quantum memory $\widetilde{O}(2^{n/3})$.
Grover instance. We find x such that $x \notin T \wedge H(x) \in H(T)$ using Grover's algorithm. The search space is the set $\{0,1\}^n$ of size 2^n, while the test function is $g(x) = 1 \iff x \notin T \wedge H(x) \in T$. Computing g requires a query to O_H and a query to O.
The number of good states is $|\{x, H(x) \in H(T) \wedge x \notin T\}|$, expected to be $|T| = 2^{n/3}$, since H is 2-to-1. Hence, this Grover instance needs $\widetilde{O}(\sqrt{2^{n-n/3}}) = \widetilde{O}(2^{n/3})$ iterations.

Limits of existing work. The practical downside of the currently available algorithms for collision is that, although they might require as little as $\widetilde{O}(2^{n/3})$ time, they would need $\widetilde{O}(2^{n/3})$ quantum memory, as in ([3]) or even sometimes $\widetilde{O}(2^{n/3})$ quantum processors as in ([21]), see also ([34]). Contrarily to the classical memory, which is cheap, *quantum memory* is a very costly part in quantum computers. It was argued by Grover and Rudolph ([34]) that a large amount of quantum memory is almost equivalent to a large amount of quantum processors. Even if one disagrees with this statement, it is widely believed that if any implementations of such algorithms will ever exist, they cannot use a large amount of quantum memory. A general discussion on the impracticality of known quantum algorithms for collision was also made by Bernstein in [9].

In summary, even if the collision problem can be solved in quantum time $\widetilde{O}(2^{n/3})$, the current algorithms require the same amount of quantum memory: the quantum time-memory product of such algorithms is $O(2^{2n/3})$, and are arguably considered impractical, even with a functioning quantum computer. The goal of our work is to present a quantum algorithm for those problems with a small number of qubits required, which will clarify the real advantage of a quantum adversary.

Quantum Algorithms for Multi-target Preimage Search The multi-target preimage search has been much less studied than quantum collisions. As said before, in the classical setting, the best known algorithm requires time $\Omega(2^{n-t})$. In the quantum setting, we present here a basic algorithm, that uses Grover search, inspired by [21]. Independently of our work, Banegas and Bernstein presented at SAC 2017 a method to perform quantum parallel multi-target preimage search ([7]). It has however little to do with the techniques studied in this paper.

Algorithm 2: Multi-target preimage search using Grover ([21])

Input. The set $T = \{y_1, \dots y_{2^t}\}$ of targets, quantum query access to the permutation $H : \{0,1\}^n \to \{0,1\}^n$ (oracle O_H).
Membership oracle. The algorithm needs a quantum oracle O_T for membership in T. A query to O_T costs quantum time $\widetilde{O}(|T|) = \widetilde{O}(2^t)$ (it can be turned to a cost in quantum memory).
Grover instance. We find x such that $H(x) \in T$ using Grover's algorithm. The search space is the set of possible plaintexts $\{0,1\}^n$ of size 2^n, while the test function is $g(x) = 1 \iff H(x) \in T$. Computing g requires a query to O_H and a query to O_T.
The number of good states is $|\{x, H(x) \in T\}|$, expected to be $|T|$, since H is a permutation. Hence, this Grover instance needs $O(\sqrt{2^{n-t}})$ iterations.

Algorithm 2 has a query complexity of $\widetilde{O}(2^{\frac{n-t}{2}})$. However, the actual time complexity can be much larger. Given a classical description of the set T, the membership oracle O_T costs either $\widetilde{O}(2^t)$ quantum memory, either $\widetilde{O}(2^t)$ quantum time. In any case, the time-memory product of this algorithm is at least $\widetilde{O}(2^t 2^{\frac{n-t}{2}}) = \widetilde{O}(2^{\frac{n+t}{2}})$. Surprisingly (and quite annoyingly), the best tradeoff would be obtained for $t = 0$, i.e one preimage only.

Comparison of our Work and Existing Work Using Different Benchmarks. The comparisons between quantum and classical time-memory products are summarized in Tables 1 and 2. Let us now consider different benchmarking scenarios and compare our work with existing work for the collision problem. When considering multiple processors in parallel, we will use the variable s, such that we have access to 2^s processors in parallel.

- If quantum memory is expensive: our quantum algorithms are the only ones that beat classical algorithms with only $O(n)$ quantum bits, with a single quantum processor. Our algorithms also beat existing quantum algorithms if we compare in terms of quantum time-space product.
- If quantum memory becomes as cheap as classical memory, but parallelization is hard then Ambainis' algorithm will have better performances than ours.

- When comparing to classical algorithms, how should we treat classical vs. quantum memory? If we consider just a time-space product (including classical space) then our single processor algorithm has a time-space product of $\widetilde{O}(2^{3n/5})$. However, if this is the quantity of interest then we can take $s = n/5$ in our quantum parallel algorithm and we will obtain a time-space product of $\widetilde{O}(2^{12n/25}) < \widetilde{O}(2^{n/2})$ which again beats the best classical algorithms with this benchmarking. If we consider that classical memory is very cheap then our algorithms compare even better to the classical ones (if we still reasonably consider the parallelization cost).

Table 1. Algorithms for collision search. The last line is valid for $s \leq n/4$.

	Time	Q-memory	C-storage	# Processors
Improved Grover search ([21])	$2^{n/3}$	$2^{n/3}$	-	$2^{n/3}$
Ambainis' algorithm ([3])	$2^{n/3}$	$2^{n/3}$	-	no
Classical parallelization ([48])	$2^{n/2-s}$	-	2^s	2^s
Our work - single processor	$2^{2n/5}$	$O(n)$	$2^{n/5}$	no
Our work - parallelization	$2^{2n/5-3s/5}$	$O(2^s n)$	$2^{n/5+s/5}$	2^s

Table 2. Algorithms for multi-target preimage search. We consider 2^s processors for the two parallelized algorithms and a single one for the rest.

	Time	Q-memory	C-storage
Classical algorithm	2^{n-t}	-	2^t
Classical algorithm - parallel	2^{n-t-s}	-	$2^t + 2^s$
Naive quantum algorithm	$2^{n/2}$	$O(n)$	-
Our work - single processor	$2^{n/2-t/6} + \min\{2^t, 2^{3n/7}\}$	$O(n)$	$\min\{2^{t/3}, 2^{n/7}\}$
Our work - parallelization	$2^{n/2-t/6-s/2} + \min\{2^t, 2^{\frac{3n-4s}{7}}\}$	$O(2^s n)$	$\min\{2^{t/3}, 2^{n/7+s/7}\}$

3.4 Cryptographic Applications of the Problems

Searching for collisions and (multi-target) preimages are recurrent generic problems in symmetric cryptanalysis. We describe here several scenarios whose security would be considerately affected by an improvement in the resolution of these problems by quantum adversaries. The improvements permitted by our algorithms will be detailed in section 6.

Hash Functions. A hash function is a function H that, given a message M of an arbitrary length, returns a value $H(M) = h$ of a fixed length n. They have many applications in computer security, as in message authentication codes, digital signatures and user authentication. Hash functions must be easy to compute. An "ideal" hash function verifies the following properties:

- Collision resistance: Finding two messages M and $M' \neq M$ such that $H(M) = H(M')$ should cost $\Omega(2^{n/2})$ by the birthday paradox [53].
- Second preimage resistance: Given a message M and its hash $H(M)$, finding another message M' such that $H(M) = H(M')$ should cost $\Theta(2^n)$ by exhaustive search. [2]
- Preimage resistance: From a hash h, finding a message M so that $H(M) = h$ should cost $\Theta(2^n)$ by exhaustive search.

We can see how, if the algorithms for solving collision search or preimages are improved, the offered security of hash functions would be reduced.

Multi-user Setting. In what follows, E_K will always denote a symmetric cipher under key K of length k, of block size n. We consider E_K as a random permutation of bit-strings $E_K : \{0,1\}^n \to \{0,1\}^n$. We consider the setting where an adversary tries to recover the keys of many users of E_K in parallel. One of the considered scenarios [13, 14, 22, 46] tries to recover one key out of the 2^t more efficiently than in the single key setting. It is easy to see how this can be associated to the multi-target preimage problem: we can for instance consider that all the 2^t users are encrypting the same message, each with a different key, and we recover the corresponding encrypted blocks. This setting seems realistic: it could be the case of users using the CTR operation mode [26], which is one of the two most popular and recommended modes (see for instance [44]), in protocols like for instance TLS [25]. The users consider $IV = 0$ and different secret keys. Recovering one key out of the 2^t would cost in a generic and classical way 2^{k-t} encryptions, for a key of length k. Similar scenarios have been studied in [29] with respect to the Even-Mansour cipher [28] and the Prince block cipher [18].

Collision Attacks on Operation Modes. Using operation modes such as CBC or CTR, block ciphers are secure up to $2^{n/2}$ encryptions with the same key [45], where collisions start to occur and reveal information about the plaintexts (see Section 2.1). The recommendation from the community is to limit the number of blocks encrypted with the same key to $\ell \ll 2^{n/2}$, but this is not always respected by standards or actual applications. Such an attack scenario is not merely theoretical, as the authors of [11] pointed out.

They proved that when the birthday bound was only weakly enforced, collision attacks were practical against 64-bit block ciphers when using CBC. In

[2] For single pipe constructions this is reduced by the blocks of length of the message M.

their scenario, the attacker lures the user into sending a great number of HTTP requests to a target website, then captures the server's replies: blocks of sensitive data encrypted under the same key. This attack has time and data complexity $O(2^{n/2})$ (practical when $n = 64$).

Bricks for Cryptanalysis Techniques. Both collision search and multi-target preimage search are often bricks used in some evolved cryptanalysis techniques, as for instance in truncated differential attacks [39] or in impossible differential attacks [38, 12] where the adversary needs to find partial output collisions to perform the attacks. Consequently, any acceleration of the algorithms solving these problems would be directly translated in an acceleration of one of the terms of the complexity, and potentially, on an improvement of the complexity of the cryptanalysis technique.

4 The Membership Oracle

In the algorithm of Brassard *et al.*, as well as in the algorithm that will be detailed in Section 5, a quantum oracle is needed for computing membership in a large, unstructured set. We formalize here this essential building block.

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

A quantum membership oracle for T is an operator O_T that computes f_T:

$$O_T(|x\rangle\,|b\rangle) = |x\rangle\,|b \oplus f_T(x)\rangle \quad .$$

The model of computation and memory. The set $T = \{x_1, \ldots, x_{2^t}\}$ for which we want to construct a quantum membership oracle is stored in some classical memory, and we require only classical access to it, meaning that for any $i \in [1, \ldots, 2^t]$, we can efficiently obtain element x_i. Notice that all x_i are distinct; this is ensured e.g by the data structure itself or by a preliminary in-place sort in $\widetilde{O}(2^t)$. We use the following quantum operations:

- A quantum creation algorithm that takes a classical input x of n bits, and n qubits initialized at $|0\rangle$ and outputs $|x\rangle$ in this register. This can be done in time n by constructing each qubit of $|x\rangle$ separately.
- A quantum unitary COMP defined as follows:

$$\forall x, y \in \{0,1\}^n, \forall b \in \{0,1\},\ COMP(|x\rangle\,|y\rangle\,|b\rangle) := |x\rangle\,|y\rangle\,|b \oplus (\delta_{xy})\rangle \quad .$$

- A quantum deletion algorithm that takes a classical input x and $|x\rangle$ and outputs $|0\rangle$. This is just done by inverting the creation algorithm.

Using those operations, we describe now how to construct O_T. We start from an input $|x\rangle\,|b\rangle$ and want to construct $|x\rangle\,|b \oplus f_T(x)\rangle$. Our construction will be clearly linear and will correspond to a quantum unitary. The idea is simple: for

each $x_i \in T$, we will check whether $x = x_i$ and update the second register accordingly.

Algorithm 3: Quantum algorithm for set membership

Start from the input $|\phi_1\rangle := |x\rangle |b\rangle$ on $n+1$ qbits. For $i = 1$ to 2^t :

- Get element x_i from T and construct a quantum register $|x_i\rangle$ using the creation operator to which we concatenate the current state $|\phi_i\rangle$.
- Apply COMP on the state $|x_i\rangle |\phi_i\rangle$.
- Discard the first register using the deletion operator. Let $|\phi_{i+1}\rangle$ be the remaining state.

The final state $|\phi_{2^t+1}\rangle$ is exactly equal to $|x\rangle |b \oplus f_T(x)\rangle$.

Proposition 1. *The above procedure implements O_T perfectly, in time $n2^t$ using $2n+1$ bits of quantum memory.*

Proof. The proof is by a straightforward induction. It is easy to see that $|\phi_{i+1}\rangle$ is the state:

$$|x\rangle |b \oplus (\delta_{xx_1}) \oplus (\delta_{xx_2}) \ldots \oplus (\delta_{xx_i})\rangle \quad .$$

By definition:

$$f_T(x) = 1 \iff x \in T \iff (x = x_1 \vee \ldots \vee x = x_{2^t})$$

which implies (all x_i are distinct):

$$\delta_{xx_1} \oplus \delta_{xx_2} \ldots \oplus \delta_{xx_i} = (x = x_1 \vee \ldots \vee x = x_i) \quad .$$

The result follows:

$$|\phi_{2^t+1}\rangle = |x\rangle \left|b \oplus \delta_{xx_1} \oplus \delta_{xx_2} \ldots \oplus \delta_{xx_{2^t}}\right\rangle = |x\rangle |b \oplus f_T(x)\rangle \quad .$$

5 Description of our Quantum Algorithms

In this section we describe our new algorithms for collision and multi-target preimage search. They use three (resp. two) instances of the amplitude amplification procedure (see Theorem 1 in Section 2).

5.1 Quantum Algorithm for Collision Finding

Our algorithm, described in Algorithm 4, relies on a balance between the cost of queries to the function and queries to the membership oracle. This balance principle was in fact already considered in [33] to improve the running time of Grover's algorithm. In the algorithm of Brassard *et al.*, when using only logarithmic quantum memory, each query costs $O(2^{n/3})$ time, so there is much room for improvement.

The way we construct the input space S_r^H and the membership oracle f_L^H allow us to decrease the projecting time while increasing the setup time. Independently from the choice of t and r, the quantum memory complexity remains $O(n)$.

Algorithm 4: Quantum algorithm for collision finding

The input is a random function $H : \{0,1\}^n \to \{0,1\}^n$ to which we have quantum oracle access. The output is a collision (x, x') such that $x \neq x'$ and $H(x) = H(x')$). The parameters r and t are fixed and will be optimized later. For $r \in [1, \ldots, n]$, let $S_r^H := \{(x, H(x)) : \exists z \in \{0,1\}^{n-r}, H(x) = \underbrace{0 \ldots 0}_{r \text{ times}} ||z\}$.

S_r^H consists of input/output pairs $(x, H(x))$ such that $H(x)$ starts with r zeros. The algorithm works as follows:

1. Construct a list L consisting of 2^{t-r} elements from S_r^H. Let $f_L^H(x) := 1$ if $\exists (x', H(x')) \in L,\ H(x) = H(x')$ and $f_L^H(x) := 0$ otherwise.
2. Apply a quantum amplification algorithm where
   - The setup is the construction of $|\phi_r\rangle := \frac{1}{\sqrt{|S_r^H|}} \sum_{x \in S_r^H} |x, H(x)\rangle$.
   - The projector is a quantum oracle query to $O_{f_L^H}$ meaning that

$$O_{f_l^H}(|x, H(x)\rangle |b\rangle) = |x, H(x)\rangle \left|b \oplus f_L^H(x)\right\rangle.$$

The above quantum amplification algorithm is essentially a Grover search algorithm for f_L^H but on input space S_r^H. The algorithm will output an element $(x, H(x))$ such that $f_L^H(x) = 1$, which means that $\exists (x', H(x')) \in L,\ H(x) = H(x')$. We will finally argue that with constant probability, $(x, H(x)) \notin L$ which will imply $x \neq x'$ and a collision for H.

Analysis of the Algorithm In this section, we make some simplifying assumptions. A more rigorous analysis including all the calculation appears in the auxiliary supplementary material. These assumptions are the following:

- The `QAA` used in our setting outputs exactly the desired state.
- $|S_r^H| \approx 2^{n-r}$.
- Let us define $Sol_f := \{x : f_L^H(x) = 1\}$. We have $|Sol_f| \approx 2 \times 2^{t-r}$. Indeed, each element of L can be mapped with its first coordinate to an element of Sol_f which corresponds to 2^{t-r} elements. Each x such that $(x, H(x)) \notin L$ is in Sol_f with probability $2^{-n+(t-r)}$. Since there are $2^n - 2^{t-r}$ such elements, this corresponds to approximately $2^{t-r} - 2^{2(t-r)-n} \approx 2^{t-r}$ elements.
- We omit all factors polynomial in n and consider that the running time of O_H is 1.

With those assumptions, we get a running time of $2^{\frac{2n}{5}}$ as we show below. If we remove all the above assumptions, we will still obtain a running time of $2^{\frac{2n}{5}}$ $(|O_H|_{RT} + O(n))$. Details of this general running time are given in the auxiliary supplementary material.

Probability of success. We constructed a list L of 2^{t-r} elements of the form $(x, H(x))$. The algorithm outputs a random $x \in Sol_f$. Our protocol succeeds if that element is not in L. Since $|L| = 2^{t-r}$ and $|Sol_f| \approx 2 \times 2^{t-r}$, we get a good outcome with probability $\frac{1}{2}$.

Time analysis. Recall that an amplification procedure $\texttt{QAA}$ uses two algorithms: a projection oracle O_P as well as a setup $\texttt{setup}$ that produces a state $|\phi\rangle = \alpha |\phi_P\rangle + \beta |\phi_P^{\perp}\rangle$.

We decompose our algorithm into four subroutines.

1. Constructing the list L: an element of L can be constructed in time $2^{r/2}$ by applying Grover's search algorithm on the function $f(x) := 1$ if $x \in S_r^H$ and $f(x) := 0$ otherwise. Since the whole list L contains 2^{t-r} elements, it can be constructed in time $2^{t-\frac{r}{2}}$.
2. Constructing $|\phi_r\rangle$: we use an algorithm $\mathcal{A} = \texttt{QAA}(\texttt{setup}_{\mathcal{A}}, \texttt{proj}_{\mathcal{A}})$, where $\texttt{setup}_{\mathcal{A}}$ builds the superposition $|\phi_0\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle$ using a query to O_H and $\texttt{proj}_{\mathcal{A}} = \sum_{x \in S_r^H} |x\rangle\langle x|$.
   $tr(P |\phi_0\rangle\langle\phi_0|) = 2^{-r}$ so we have to perform $2^{r/2}$ iterations, *i.e.* make $2^{r/2}$ calls to $\texttt{setup}_{\mathcal{A}}$ and $\texttt{proj}_{\mathcal{A}}$. Algorithm $\mathcal{A}$ takes therefore time $2^{r/2}$.
3. Constructing $O_{f_L^H}$. The details of this construction appear in Section 4. In particular, we saw that $O_{f_L^H}$ runs in time 2^{t-r} by testing sequentially against the elements of L (recall we dismissed the factor n for simplicity).
4. Performing the main amplitude amplification: Algorithm $\mathcal{B} = \texttt{QAA}(\texttt{setup}_{\mathcal{B}} = \mathcal{A}, \texttt{proj}_{\mathcal{B}})$, where $\mathcal{A}$ is the setup routine that constructs state $|\phi_r\rangle$, and $\texttt{proj}_{\mathcal{B}} = \sum_{x: f_L^H(x)=1} |x\rangle\langle x|$. $O_{\texttt{proj}_{\mathcal{B}}}$ can be done with 2 calls to $O_{f_L^H}$.

The probability that a random $x \in S_r^H$ satisfies $f_L^H(x) = 1$ is $\frac{|Sol_f|}{|S_r|} = \frac{2*2^{t-r}}{2^{n-r}}$ so $tr(\texttt{proj}_{\mathcal{B}} |\phi_r\rangle\langle\phi_r|) = \frac{2*2^{t-r}}{2^{n-r}} = 2^{-n+t+1}$ and Algorithm $\mathcal{B}$ makes $2^{\frac{n-t-1}{2}}$ calls to $\mathcal{A}$ and $O_{f_L^H}$. As a result, algorithm $\mathcal{B}$ runs in time:

$$2^{\frac{n-t-1}{2}} (\texttt{setup}_{\mathcal{B}} + \texttt{proj}_{\mathcal{B}}) = 2^{\frac{n-t-1}{2}} \left(2^{r/2} + 2^{t-r}\right) .$$

The running time of the procedure is the time to create the list L plus the time to run algorithm $\mathcal{B}$, which is

$$2^{\frac{n-t-1}{2}} \left(2^{r/2} + 2^{t-r}\right) + 2^{t-\frac{r}{2}} .$$

A quick optimization of the above expression imposes $t = \frac{3n}{5}$ and $r = \frac{2t}{3} = \frac{2n}{5}$. This realizes a balance in $\mathcal{B}$ between the cost of the setup and the cost of a projection, and between the cost of $\mathcal{B}$ and the cost of computing L.

This gives a total running time of $2^{\frac{2n}{5}}$, up to a small multiplicative factor in n.

Memory analysis. The quantum amplitude amplification algorithms and the circuit $O_{f_L^H}$ only require quantum circuits of size $O(n)$: the quantum memory (number of qubits) needed is low. As for the classical memory required, the only data we need to store is the list L that contains $2^{t-r} = 2^{\frac{n}{5}}$ elements.

Theorem 2. *Let $H : \{0,1\}^n \to \{0,1\}^n$ be a random function computable efficiently. There exists a quantum algorithm running in time $\widetilde{O}\left(2^{\frac{2n}{5}}\right)$, using $\widetilde{O}\left(2^{\frac{n}{5}}\right)$ classical memory and $O(n)$ quantum space, that outputs a collision of H.*

5.2 Quantum Algorithm for Multi-target Preimage Search

Here, we are given a function H and a list $L' = \{y_1, \ldots, y_{2^t}\}$ of elements of size 2^t. The goal is to find x such that $\exists y_i, H(x) = y_i$, the preimage of one of them. The algorithm used is very similar to Algorithm 4.

Algorithm 5: Quantum algorithm for multi-target preimage search

The input is a random permutation $H : \{0,1\}^n \to \{0,1\}^n$ to which we have quantum oracle access, and a list $L' = \{y_1, \ldots, y_{2^t}\}$. The output is the preimage of one of the y_i. For $r \in [1, \ldots, n]$, let $S_r^H := \{x : \exists z \in \{0,1\}^{n-r}, H(x) = \underbrace{0 \ldots 0}_{r \text{ times}} || z\}$. S_r^H consists of elements x such that $H(x)$ starts with r zeros. The algorithm works as follows:

1. Construct a list L consisting of all elements of L' that start with r zeros. L contains 2^{t-r} elements on average (and the deviation is actually small). Let $f_L^H(x) := 1$ if $H(x) \in L$ and $f_L^H(x) := 0$ otherwise.
2. Apply a quantum amplification algorithm where
   - The setup is the construction of $|\phi_r\rangle := \frac{1}{\sqrt{|S_r^H|}} \sum_{x \in S_r^H} |x\rangle$.
   - The projector is a quantum oracle query to $O_{f_L^H}$ meaning that

$$O_{f_l^H}(|x\rangle |b\rangle) = |x\rangle \left| b \oplus f_L^H(x) \right\rangle.$$

The above quantum amplification algorithm is essentially a Grover search algorithm for f_L^H but on input space S_r^H. The algorithm will output an element x such that $f_L^H(x) = 1$, which means that $H(x) \in L$. Notice that we slightly changed the definitions here. In particular, we didn't need to keep the couples $(x, H(x))$ and we could just work on x.

The only difference with respect to the previous algorithm is that the list L' of targets has to be read, even in an online manner, to create the sublist L. This operation will take time 2^t.

Because the rest of the algorithm remains unchanged, the total running time is:

$$2^{\frac{n-t}{2}} \left(2^{\frac{r}{2}} + 2^{t-r}\right) + 2^t$$

which is minimized for $r = \frac{2t}{3}$ and $t = \frac{3n}{7}$. We distinguish 2 cases:

- if $t \leq \frac{3n}{7}$, we take $r = \frac{2t}{3}$ and the above running time becomes

$$2^{n/2-t/6} + 2^t \leq 2^{n/2-t/6+1}.$$

- if $t \geq \frac{3n}{7}$, we truncate the list L' beforehand so that it has $2^{3n/7}$ elements and we apply our algorithm on this list. The running time will therefore be $2^{3n/7}$.

Memory analysis. The only data we need to store is the list L that contains $2^{t-r} = 2^{\frac{t}{3}}$ elements. The reason why we do not have to store all elements of L' is that we can discard all elements of L' that are not in S_r^H as soon as we receive them and locally (L' is analyzed in an online way). The quantum algorithm is still a circuit of size $O(n)$, without external quantum memory.

Theorem 3. *Let $H : \{0,1\}^n \to \{0,1\}^n$ be a random permutation. Given a list of 2^t elements, with $t \leq \frac{3n}{7}$, there exists a quantum algorithm running in time $\widetilde{O}\left(2^{n/2-t/6}\right)$, using $\widetilde{O}\left(2^{\frac{t}{3}}\right)$ classical memory and $O(n)$ quantum space, that finds the preimage of one of them.*

Theorem 4. *Let $H : \{0,1\}^n \to \{0,1\}^n$ be a random permutation. Given a list of $2^{\frac{3n}{7}}$ elements, there exists a quantum algorithm running in time $\widetilde{O}\left(2^{\frac{3n}{7}}\right)$, using $\widetilde{O}\left(2^{\frac{n}{7}}\right)$ classical memory and $O(n)$ quantum space, that finds the preimage of one of them.*

A similar analysis can be done with only marginal differences if we replace the random permutation by a random function.

5.3 Parallelization and Time-space tradeoff

Assume that the adversary has now 2^s registers of n qubits available. A simple way to trade space (more qubits) for time is to run in parallel multiple instances of the algorithm. We call this process *outer parallelization*, and emphasize that *quantum memory* corresponds to the number of quantum processors working in parallel.

List computation. In the case of collision search, computing the list L now costs only $2^{t-r/2-s}$ time. Notice, however, that the number of queries to O_H remains $2^{t-r/2}$.

Outer parallelization. Our algorithm consists of iterations of an operator that amplifies the amplitude of the good states (recall that $2^{\frac{n-t}{2}}$ such iterations are performed). So, instead of running only one instance and getting a good result with probability close to 1, we can run multiple instances in parallel with less iterations for each. The number of queries made to O_H will be the same.

By running $O(2^s)$ instances, to ensure success probability $O(1)$, we need each of them to have success probability $O(2^{-s})$. So instead of running $2^{\frac{n-t}{2}}$ iterations of the outer amplification procedure, only $2^{\frac{n-t-s}{2}}$ suffice. The running time for collision becomes

$$2^{\frac{n-t-s}{2}}\left(2^{r/2} + 2^{t-r}\right) + 2^{t-r/2-s} \ .$$

In collision search, this is $t = \frac{3n}{5} + \frac{3s}{5}$ which gives $r = \frac{2n}{5} + \frac{2s}{5}$, a classical memory $t - r = \frac{n}{5} + \frac{s}{5}$ and a time complexity exponent $t - \frac{r}{2} - s = \frac{2n}{5} - \frac{3s}{5}$.

In order for those parameters to be valid for collision, we need $n - t - s \geq 0$ with $t = \frac{3n}{5} + \frac{3s}{5}$ which gives $s \leq \frac{n}{4}$.

For multi-target preimage, the running time becomes

$$2^{\frac{n-t-s}{2}} \left(2^{r/2} + 2^{t-r}\right) + 2^{t-s} \ .$$

The optimal value of r is still $r = \frac{2}{3}t$. In multi-target preimage search, the optimal value of t is achieved for $\frac{n}{2} - \frac{t}{6} - \frac{s}{2} = t - s$ or equivalently $t = \frac{3n}{7} + \frac{3s}{7}$. The running time becomes $2^{3n/7-4s/7}$ and the used classical memory is $2^{\frac{n+s}{7}}$.

Theorem 5 (Outer parallelization). *Let $H : \{0,1\}^n \to \{0,1\}^n$ be a random permutation. Given a list of 2^t elements, with $t \leq \frac{3n+3s}{7}$, there exists a quantum algorithm with 2^s quantum processors running in time $\widetilde{O}\left(2^{n/2-t/6-s/2}\right)$, using $\widetilde{O}\left(2^{\frac{t}{3}}\right)$ classical memory, that finds the preimage of one of them.*

Theorem 6 (Outer parallelization). *Let $H : \{0,1\}^n \to \{0,1\}^n$ be a random permutation. Given a list of 2^t elements, with $t = \frac{3n+3s}{7}$, there exists a quantum algorithm with 2^s quantum processors running in time $\widetilde{O}\left(2^{3n/7-4s/7}\right)$, using $\widetilde{O}\left(2^{\frac{n+s}{7}}\right)$ classical memory, that finds the preimage of one of them.*

There is a range of values of s where the time-space product is effectively smaller than in the classical setting (where all current algorithms obtain an exponent $\frac{n}{2}$). The limit value is $s = \frac{n}{6}$ for preimage search and $s = \frac{n}{4}$ for collisions.

Inner parallelization. It is also possible to parallelize computations in the algorithm itself, especially its most costly building block: the membership oracle $O_{f_L^H}$. We studied this and concluded that this way of parallelizing is not as efficient as outer parallelization.

5.4 Accurate Computations and Parameters

In what precedes, we didn't take into account four sources of possible differences between theory and practice. First, hidden constants: we dismiss the $\pi/4$ factor that stems from amplitude amplification. Second, the logarithmic factor n that appears in the membership oracle. Third, the errors that propagate in the amplitude amplification procedure. Fourth, the cost of a query to the oracle O_H. This last one is actually the most relevant parameter.

Let 2^c be the time complexity of a query. We adapt the parameters r and t as follows:

- In any case, $r = \frac{2}{3}(t - c + \ln_2(n))$ balances the costs;
- In multi-target preimage search, $t = \frac{3n}{7} + \frac{4c}{7} + \frac{2\ln_2(n)}{7}$ is the new complexity exponent. Notice that our method also amortizes the cost of O_H w.r.t a simple Grover search.

- In collision search, $t = \frac{3n}{5} - \frac{4c}{5} + \frac{4\ln_2(n)}{5}$ and time complexity exponent is $\frac{2n}{5} + \frac{4c}{5} + \frac{\ln_2(n)}{5}$.

These computations mean that there is no surprise: the n factor missing above does no more than multiplying the time complexity by 4 ($n = 128$) or 16 ($n = 256$), and by taking into account the cost of a query 2^c, the time complexity does not exceed the previous one multiplied by 2^c. It even behaves better.

In tables 4 and 3, we give some complexity results *without* taking into account the n and 2^c factors. We do *not* take into account *ancilla qubits*, i.e additional qubits used during the computation. Detailed studies on the quantum cost of implementing Grover's algorithm have been made, e.g in [31] for an AES exhaustive key search and [4] for preimage search on SHA-2 and SHA-3 using Grover. Due to space constraints, we cannot go into the technicalities of quantum implementations and restrict ourselves to high-level comparisons ; notice, however, that the two aforementioned articles could help in deriving precise hardware costs for our algorithms.

Errors that Propagate in the Amplitude Amplification Procedure. We perform many instances of `QAA` in our algorithm so it is important to understand how the errors propagate and see if it doesn't create a large cost in the algorithm. In the additional material (A), we perform a detailed analysis of what is happening but we want to briefly describe here the behavior of those errors in our algorithm. Let us consider our first `QAA` algorithm to construct $|\phi_r\rangle$. There are 2 factors that can induce errors here: (1) the fact that we do not know exactly $|S_r^H|$ and (2) the fact that even with perfect knowledge of the angle used in `QAA`, the algorithm will construct a state close to $|\phi_r\rangle$ but can't hit it exactly. The second problem is solved by using a construction from [20]. In order to solve the first problem, we will use the fact that H is random so that the uncertainty will remain very small. To give a rough idea, the first `QAA` will give a state $|\phi_{output}\rangle$ such that

$$|\langle\phi_{output}|\phi_r\rangle| \geq \cos(2^{r/2-n/2} + o(2^{r/2-n/2})) \ .$$

This error will then propagate to the second `QAA`. We have two possible scenarios:

- For the collision problem, we have $r = 2n/5$ and we repeat the second `QAA` $2^{n/5}$ times. The error in the angle will increase by this factor so the final error will be $\approx 2^{n/5}2^{r/2-n/2} \ll 1$.
- For the preimage problem, we have $r = 2n/7$ and we repeat the second `QAA` $2^{2n/7}$ times. The error in the angle will increase by this factor so the final error will be $\approx 2^{2n/7}2^{r/2-n/2} \ll 1$.

This means that the final probability of success will be reduced only marginally.

Table 3. Quantum collision attack – rounded exponents

n	Space (registers)	Classical memory	Quantum time comp.	Quantum time-space prod.	Classical time-space prod.
128	$O(1)(s=0)$	26	51	51	64
128	$s=n/6=21$	30	39	60	64
256	$O(1)(s=0)$	51	102	102	128
256	$s=n/6=43$	60	77	119	128

Table 4. Quantum multi-target preimage attack – rounded exponents

n	Space (registers)	Targets	Classical memory	Quantum time comp.	Quantum time-space prod.	Classical time comp.
128	$O(1)(s=0)$	55	18	55	55	73
128	$s=n/8=16$	62	21	47	63	66
256	$O(1)(s=0)$	110	37	110	110	146
256	$s=n/8=32$	124	41	92	124	132

5.5 Many Collisions

For some purposes, it happens that we do not want to retrieve only one collision pair, but many of them. Suppose 2^c are needed. We modify the parameters in our algorithm to take this into account: now $t = 3n/5 + 6c/5$ and $r = 2n/5 + 4c/5$. Each call returns a collision involving one element of the arbitrary list L of size 2^{t-r}. Hence, we expect 2^{t-r} such collisions to be found by repeating our algorithm and sharing the list L: this forces $t - r > c \Rightarrow c < \frac{n}{3}$. Outside this range, c constraints the size of L: we must have $t - r = c$, $t = 3c$, computing L now costs $2^{t-(t-c)/2} = 2^{2c}$ and the list has 2^c elements. The time complexity exponent becomes $\frac{n}{2} + \frac{c}{2}$; it still presents an advantage over classical collision search.

Theorem 7 (Searching many collisions.). *Given a random function $H : \{0,1\}^n \to \{0,1\}^n$ on n bits, there exists a quantum algorithm using $O(n)$ qubits and outputting 2^c collisions:*

- *If $c < \frac{n}{3}$, in time $\widetilde{O}\left(2^{2n/5+4c/5}\right)$, using $2^{n/5+2c/5}$ classical memory;*
- *If $c > \frac{n}{3}$, in time $\widetilde{O}\left(2^{n/2+c/2}\right)$, using 2^c classical memory.*

To ensure that the collisions found are all distinct, one should also multiply this requirements by a small (logarithmic) factor.

6 Impact on Symmetric Cryptography

We discuss below the applications of our new algorithms on the cryptographic scenarios detailed in section 3.4.

We ask the reader to keep in mind that these results seemed particularly surprising as it was conjectured that quantum algorithms for solving these problems wouldn't be more efficient than classical ones.

6.1 On Hash Functions

We consider the setting presented in section 3.4: finding collisions and multi-target preimages on hash functions in a post-quantum world can be considerably accelerated by using the new algorithms. It is important to point out that this can be done considering the Q1 setting for the attacker described in Section 2.2: that is, just having access to a quantum computer will allow her to accelerate the attack, and she has no need of access to quantum encryption oracles with superposition queries.

To correspond precisely to the description of the problem, we can consider messages with the same length as the hash value.[3] Indeed, to find a collision, the attacker just has to provide the hash function itself as input for Algorithm 4 (Section 5.1). Algorithm 4 will output a collision with a complexity of $\widetilde{O}(2^{2n/5})$ in time and queries, using $\widetilde{O}(2^{n/5})$ classical memory and $O(n)$ qubits. This is to compare with the previous best time complexity of $O(2^{n/2})$.

Finding a preimage out of a set of 2^t generated hash values can be done with Algorithm 5 from Section 5.2. It is optimal for $t = 3n/7$ with a cost of $\widetilde{O}(2^{3n/7})$ in time and queries, using $\widetilde{O}(2^{n/7})$ classical memory and $O(n)$ qubits. This should be compared to a classical time complexity of $2^{n-t} = 2^{4n/7}$, or to the previous best quantum attack in $2^{n/2}$, ours being the most performant one. Tables 3 and 4 give concrete values when the time-space tradeoff is used.

6.2 On the Multi-User Setting

The scenario that we presented in section 3.4 can also be accelerated by Algorithm 5 of Section 5.2. In this case, the attacker recovers a list of ciphertexts generated from the same plaintext, each encrypted under a different key on size k (one key per user).

The goal is to recover one key out of the total 2^t. In this case, we can consider the attacker scenario Q1: we do not need access to a quantum *encryption* oracle, but instead implement the function that encrypts a fixed plaintext under the key in argument (as we would do for an exhaustive search with a Grover attack). In this case though, the target ciphertexts must be recovered classically. When the key has the same size as the ciphertext, we can directly apply the multi-target preimage search algorithm, that will be optimal for a value of $2^t = 2^{3k/7}$ users. The best time complexity we can achieve here is $\widetilde{O}(2^{3k/7})$, compared to the previous best classical $O(2^{4k/7})$ and the previous best quantum $\widetilde{O}(2^{k/2})$.

Bigger Key than the State. If the key is bigger than the ciphertext, i.e. $k = mn$, we re-construct the problem solved by Algorithm 5 by considering that each user encrypts not one, but m fixed plaintexts.

[3] Some blocks fixed to a random value can be considered previously for randomization.

Less multi-users than optimal. If the number of multiusers is smaller than $2^{3k/7}$, we will obtain less gain in the complexities, but still considerable with respect to previous attacks. We can consider, to illustrate this, the attack in [29] presented at Asiacrypt 2014 on the Prince block cipher [18]: In this attack, the authors proposed a technique that provided improved complexity regarding already the best known previously classical multi-target preimage attacks, and they were able to recover one key of size 128 bits out of 2^{32} in time 2^{65} (already improved with respect to the naive $2^{128-32} = 2^{96}$ given by the best generic algorithm). If we apply in this case our algorithm we recover a time complexity of

$$2^{\frac{k}{2}-\frac{t}{6}} + 2^t = 2^{\frac{128}{2}-\frac{32}{6}} + 2^3 2 = 2^{64-5.33} = 2^{58.67},$$

which improves upon previous results. Our results only need a classical memory of $2^{18.3}$ and $O(n)$ qubits (compared to a memory need of 2^{32}). Parallelization can also reduce the time complexity in this scenario, but for the sake of simplicity, we won't go into the details and remit to Section 5.3.

6.3 On Operation Modes

As a quantum adversary, we can improve the classical collision attack on CBC introduced in Section 3.4 with the help of our algorithm from Section 5.2. In this scenario, the attacker has to be placed in the Q2 model from Section 2.2: she has access to 2^t classically encrypted blocks, to a quantum computing device and also quantum encryption oracle using the same secret key K.[4] After recovering the 2^t ciphertexts that form the list $L' = C_1, \ldots, C_{2^t}$, we try to find a preimage x of one of them, i.e., find x such that $E_K(x) = C_i$ for $i \in \{1, \ldots, 2^t\}$. This can be done by directly applying Algorithm 5.

Once we find such an x, we can recover P_i, i.e. the plaintext that generates C_i through encryption. Due to the CBC construction, we know that $E_K(P_i \oplus C_{i-1}) = C_i$. Therefore, and as C_{i-1} is known for the attacker, if we recover $x = P_i \oplus C_{i-1}$, we also recover P_i. This can be done by a quantum adversary with a cost in time of $\widetilde{O}(2^{3n/7})$, compared to the classical $O(2^{n/2})$.

In Section 7 we discuss the impact this attack should have on maximal data length enforcement.

Frequent rekeying. If we consider a scenario where the user could be forced to change his key after a few encryptions, this previous attack could be translated in a key-recovery one, in the Q1 model, with a similar procedure as in the multi-user case. We first recover classically 2^t ciphertexts, generated by the encryption of one plaintext with 2^t different keys, and next search for a preimage of one of these multitargets.

6.4 On Bricks for Cryptanalysis Techniques

The last scenario proposed in Section 3.4 is less concrete but of great importance. Being very general, the algorithms that we presented here may be used as

[4] See Section 2.2 for a justification of the Q2 model.

building blocks by cryptanalysts. With powerful black-box tools and available trade-offs, quantized classical cryptanalysis might become indeed more efficient.

Let us consider as an example the analysis of quantum truncated differential distinguishers from [37]. The aim of the attack is to find a pair of plaintexts with a difference belonging to a certain set Δ_{in}, that generate a pair of ciphertexts belonging to another particular set of differences Δ_{out}, which is equivalent to colliding in a part of the output state. The attack succeeds if such a pair is found quicker than for a random permutation. The probability of this happening for the attack cipher is denoted by 2^{-h_T}.

We consider the case where a single structure[5] is enough for finding the good pair statistically, i.e. if $2^{h_T} \leq 2^{2|\Delta_{in}|-1}$. The authors remark that finding the good pair will cost $O(2^{h_T/3})$ queries for a quantum adversary. But this would also cost the same amount in space. We could, instead, apply our new algorithm, allowing the quantum space needed to remain polynomial in n with a time complexity still improved over the classic one.

7 Conclusion

7.1 Efficient Algorithms for Collision and Multi-Target Preimage Search

We have presented a quantum algorithm for collision and preimage search, based on the amplitude amplification procedure, that is sub-optimal in terms of query complexity but beats the current best algorithm in terms of time complexity with small quantum memory.

To the best of our knowledge, this is the first generic procedure that solves this problem effectively faster than classically, when only linear quantum space is available. Our algorithm can also be parallelized, providing better performance than classical algorithms for a wide range or parameters.

7.2 Impact on Symmetric Primitives

From the applications presented in Section 6, we can obtain the following conclusions:

Open Problem on Best Quantum Multi-target Preimages. As we already pointed out, in [9], the author remarked that all post-quantum algorithms for collision search had an *effective* cost of at least $2^{n/2}$. In Eurocrypt 2015 [10], section 3.2, the authors further notice that the post-quantum cost for multi-target preimage attacks is also of $2^{n/2}$, and they provide the following example: for $n = 256$ and $2^t = 2^{56}$, they claim that the best quantum algorithm has a cost of 2^{128}, though

[5] A structure is a set of plaintexts of size $2^{|\Delta_{in}|}$ belonging to the same truncated difference Δ_{in}, which means that they allow to build $2^{2|\Delta_{in}|-1}$ pairs.

it only needs 2^{100} queries. With our algorithms, this implication does not hold anymore: it is possible to attack their example with a time complexity of

$$2^{100}(2^{56/3} + 2^{56/3}) + 2^{56} \approx 2^{119.6}$$

by applying Algorithm 5, which is clearly better than the classical attack, and using a polynomial amount of qubits.

On Maximal Data Length to Enforce. While attacking operation modes via collisions, 2^t data is recovered classically. This 2^t can be significantly smaller than $2^{n/2}$, and the attack would still have an advantage over the birthday paradox. In fact, when more data is available, the time complexity of the quantum computations decreases up to the limit $\widetilde{O}\left(2^{5n/12}\right)$ (when $t = n/2$).

We can forget about the term 2^t, as we are considering $t < n/2$, and the quantum procedure has complexity $\widetilde{O}\left(2^{\frac{n}{2}-\frac{t}{6}}\right)$, which offers a factor $2^{-t/6}$ compared to classical collision search, independent of the block size n. The security requirements will determine the maximal amount of data that can be generated with a given key.

Is doubling the key length enough? The multi-user scenario, as well as the re-keying one make us wonder about the actual security offered by symmetric ciphers in a post-quantum world. By accelerating collision search, we showed that Grover's exhaustive key search is not the only general threat posed to them: the block size is also a relevant parameter in quantum attacks.

These results increase our impression that many scenarios and settings should be carefully studied before being able to promise certain security levels in a post-quantum world.

7.3 Open Problems And Future Work.

Our result fills a gap that existed between the theoretical query complexity of collision search and the actual requirements of an attack. It follows recent non-intuitive results in quantum symmetric cryptanalysis (see e.g [36]), that have shown the necessity of a better understanding of how symmetric primitives could be affected by quantum adversaries. To our opinion, many such counter-intuitive results are yet to appear.

This work reopens the direction of designing improved quantum algorithms for collisions and preimage finding when the quantum computer does not have access to an exponential amount of quantum memory. The algorithm we propose will not be dismissed as implausible if we want to prove security against quantum attackers: the quantum memory needed is reasonably small. We have been able to propose several significant complexity improvements thanks to this result. Although $2n/5$ is the optimal exponent of our collision algorithm, it introduces additional structure (a prefix of the image is chosen) that is not relevant in many applications: is it possible to get rid of these specificities and bring the exponent down to $n/3$?

Acknowledgements. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n^o 714294 - acronym QUASYModo).

References

1. Aaronson, S., Shi, Y.: Quantum lower bounds for the collision and the element distinctness problems. J. ACM 51(4), 595–605 (2004)
2. Ambainis, A.: Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. Theory of Computing 1(1), 37–46 (2005), http://dx.doi.org/10.4086/toc.2005.v001a003
3. Ambainis, A.: Quantum Walk Algorithm for Element Distinctness. SIAM J. Comput. 37(1), 210–239 (2007), http://dx.doi.org/10.1137/S0097539705447311
4. Amy, Matthew, Di Matteo, Olivia, Gheorghiu, Vlad, Mosca, Michele, Parent, Alex, Schanck, John M.: Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3. In: IACR Cryptology ePrint Archive, 2016, p. 992
5. Anand, M.V., Targhi, E.E., Tabia, G.N., Unruh, D.: Post-quantum security of the cbc, cfb, ofb, ctr, and XTS modes of operation. In: PQCrypto. Lecture Notes in Computer Science, vol. 9606, pp. 44–63. Springer (2016)
6. Andreeva, E., Bouillaguet, C., Fouque, P., Hoch, J.J., Kelsey, J., Shamir, A., Zimmer, S.: Second preimage attacks on dithered hash functions. In: Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4965, pp. 270–288. Springer (2008)
7. Banegas, Gustavo, Bernstein, Daniel J.: Low-communication parallel quantum multi-target preimage search. In: SAC 2017
8. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A Concrete Security Treatment of Symmetric Encryption. In: FOCS. pp. 394–403. IEEE Computer Society (1997)
9. Bernstein, D.J.: Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? SHARCS'09 Special-purpose Hardware for Attacking Cryptographic Systems p. 105 (2009)
10. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: 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. Lecture Notes in Computer Science, vol. 9056, pp. 368–397. Springer (2015)
11. Bhargavan, K., Leurent, G.: On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. IACR Cryptology ePrint Archive 2016, 798 (2016), http://eprint.iacr.org/2016/798
12. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer (1999)
13. Biham, E.: How to decrypt or even substitute des-encrypted messages in 2^{28} steps. Inf. Process. Lett. 84(3), 117–124 (2002), http://dx.doi.org/10.1016/S0020-0190(02)00269-7

14. Biryukov, A., Mukhopadhyay, S., Sarkar, P.: Improved time-memory trade-offs with multiple data. In: Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3897, pp. 110–127. Springer (2006)
15. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random Oracles in a Quantum World. In: Lee, D., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011, Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-25385-0_3
16. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011), http://dx.doi.org/10.1007/978-3-642-25385-0_3
17. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 8043, pp. 361–379. Springer (2013)
18. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer (2012)
19. Brassard, G., Høyer, P., Kalach, K., Kaplan, M., Laplante, S., Salvail, L.: Merkle puzzles in a quantum world. In: Advances in Cryptology–CRYPTO 2011, pp. 391–410. Springer (2011)
20. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics 305, 53–74 (2002)
21. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: LATIN. Lecture Notes in Computer Science, vol. 1380, pp. 163–169. Springer (1998)
22. Chatterjee, S., Menezes, A., Sarkar, P.: Another look at tightness. In: Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7118, pp. 293–319. Springer (2012)
23. Damgård, I., Funder, J., Nielsen, J.B., Salvail, L.: Superposition Attacks on Cryptographic Protocols. In: Padró, C. (ed.) Information Theoretic Security - 7th International Conference, ICITS 2013, Singapore, November 28-30, 2013, Proceedings. Lecture Notes in Computer Science, vol. 8317, pp. 142–161. Springer (2013), http://dx.doi.org/10.1007/978-3-319-04268-8_9
24. De Wolf, R.: Quantum Computing: Lecture Notes (2013)
25. Dierks, T., Rescorla, E.: The transport layer security (tls) protocol version 1.2. In: IETF RFC 5246 (2008)
26. Diffie, W., Hellman, M.: Privacy and authentication: An introduction to cryptography. In: Proceedings of the IEEE. p. 397427. 67 (1979)
27. Ehrsam, W.R., Meyer, C.H., Smith, J.L., Tuchman, W.L.: Message verification and transmission error detection by block chaining. US Patent 4074066 (1976)

28. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptology 10(3), 151–162 (1997), http://dx.doi.org/10.1007/s001459900025
29. Fouque, P., Joux, A., Mavromati, C.: Multi-user collisions: Applications to discrete logarithm, even-mansour and PRINCE. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 420–438. Springer (2014)
30. Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: Annual Cryptology Conference. pp. 60–89. Springer (2016)
31. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying Grover's Algorithm to AES: Quantum Resource Estimates. In: PQCrypto. Lecture Notes in Computer Science, vol. 9606, pp. 29–43. Springer (2016)
32. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996), http://doi.acm.org/10.1145/237814.237866
33. Grover, L.K.: Trade-offs in the quantum search algorithm. In: Physical Review A, vol 66 (2002)
34. Grover, L.K., Rudolph, T.: How significant are the known collision and element distinctness quantum algorithms? Quantum Information & Computation 4(3), 201–206 (2004), http://portal.acm.org/citation.cfm?id=2011622
35. Kaplan, M.: Quantum attacks against iterated block ciphers. CoRR abs/1410.1434 (2014), http://arxiv.org/abs/1410.1434
36. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: CRYPTO (2). Lecture Notes in Computer Science, vol. 9815, pp. 207–237. Springer (2016)
37. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symmetric Cryptol. 2016(1), 71–94 (2016), http://tosc.iacr.org/index.php/ToSC/article/view/536
38. Knudsen, L.R.: DEAL – A 128-bit cipher. Technical Report, Department of Informatics, University of Bergen, Norway (1998)
39. Knudsen, L.R.: Truncated and higher order differentials. In: Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings. Lecture Notes in Computer Science, vol. 1008, pp. 196–211. Springer (1994)
40. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: 2011. LNCS, vol. 6733, pp. 306–327. Springer (2011)
41. Kutin, S.: Quantum lower bound for the collision problem with small range. Theory of Computing 1(2), 29–36 (2005), http://www.theoryofcomputing.org/articles/v001a002
42. Kuwakado, H., Morii, M.: 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. pp. 2682–2685. IEEE (2010), http://dx.doi.org/10.1109/ISIT.2010.5513654
43. Kuwakado, H., Morii, M.: 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. pp. 312–316. IEEE (2012), http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6400943

44. Lipmaa, H., Rogaway, P., Wagner, D.: Comments to nist concerning aes modes of operations: Ctr-mode encryption (2000), http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ctr/ctr-spec.pdf
45. McGrew, D.A.: Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. IACR Cryptology ePrint Archive 2012, 623 (2012), http://eprint.iacr.org/2012/623
46. Menezes, A.: Another look at provable security. In: Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, p. 8. Springer (2012)
47. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
48. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with application to hash functions and discrete logarithms. In: CCS '94, Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 2-4, 1994. pp. 210–218. ACM (1994)
49. Pollard, J.M.: A monte carlo method for factorization. BIT Numerical Mathematics 15(3), 331–334 (1975), http://dx.doi.org/10.1007/BF01933667
50. Shor, P.W.: 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. pp. 124–134. IEEE Computer Society (1994), http://dx.doi.org/10.1109/SFCS.1994.365700
51. Simon, D.R.: On the Power of Quantum Cryptography. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 116–123. IEEE Computer Society (1994), http://dx.doi.org/10.1109/SFCS.1994.365701
52. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Eurocrypt 2015. vol. 9057, pp. 755–784. Springer (2015), preprint on IACR ePrint 2014/587
53. Yuval, G.: How to swindle rabin. Cryptologia 3(3), 187–191 (1979), http://dx.doi.org/10.1080/0161-117991854025
54. Zhandry, M.: How to construct quantum random functions. In: FOCS. pp. 679–687. IEEE Computer Society (2012)
55. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Info. Comput. 15(7-8), 557–567 (May 2015), http://dl.acm.org/citation.cfm?id=2871411.2871413
56. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. International Journal of Quantum Information 13(04), 1550014 (2015)

A Auxiliary Supplementary Material: Errors in the Amplitude Amplification

A.1 More Precise Statements for QAA Algorithms with Errors

We first restate a more formal statement for quantum amplitude amplification.

Theorem 8 ([20], Quantum amplitude amplification, with known angle). *Let P be a projector acting on n qubits and O_P a projection oracle for P. Let $\mathcal{A}$ be a quantum unitary that produces a state $|\phi\rangle = \alpha |\phi_P\rangle + \beta \left|\phi_P^{\perp}\right\rangle$ where $|\phi_P\rangle \in Im(P)$ and $\left|\phi_P^{\perp}\right\rangle \in Ker(P)$. Suppose we know $|\alpha| = \sin(\theta)$ for some $\theta \in [0, \pi/2]$. There exists a quantum unitary circuit that:*

- *Consists of exclusively N calls to $O_P, O_P^{\dagger}, \mathcal{A}, \mathcal{A}^{\dagger}$.*
- *Produces on input $|0\rangle$ a quantum state $|\phi_{output}\rangle$ satisfying*

$$|\langle \phi_{output} | \phi_P \rangle| = \sin(2N\theta + \theta).$$

The algorithm $\mathcal{A}$ is called the setup and the projection P the projector of the quantum amplification algorithm. This whole procedure will be denoted

$$\texttt{QAA}(\texttt{setup}, \texttt{proj}) = \texttt{QAA}(\mathcal{A}, P)$$

and its running time is

$$N\left(|\mathcal{A}|_{RT} + |O_P|_{RT}\right).$$

where the notation $|\cdot|_{RT}$ represents the running time of the respective algorithms. By taking $N = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rfloor$ we obtain a state $|\phi_{output}\rangle$ satisfying

$$|\langle \phi_{output} | \phi_P \rangle| = \sin\left(\frac{\pi}{2} - \delta\right) \approx 1 - \frac{\delta^2}{2} \quad \text{with } \delta \leq 2\theta.$$

The error δ achieved can be quite small, especially because we consider very small values for θ. However, in our collision and multi-target preimage algorithms, we will use amplitude amplification a large number of times, so we want to make sure each instantiation has a very high precision.

We will have to deal with two problems. First, the error has to be reduced to something smaller than 2θ; second, there is in our case a small uncertainty on the value θ: we have to make sure that it does not impact the precision.

Reducing the error. In [20], the authors present a way to reduce the error. After the $N = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rfloor$ steps of the procedure, they apply an additional step which is almost the same, but using two extra phase flips Z_γ and $Z_{\gamma'}$ for some well chosen angles γ, γ'. The phase flip is the following operation:

$$Z_\gamma(|0\rangle) = |0\rangle \quad ; \quad Z_\gamma(|1\rangle) = e^{i2\pi\gamma} |1\rangle .$$

If we are able to choose any angles $\gamma, \gamma' \in \mathbb{R}$, then [20] shows that we can perform the amplitude amplification without any errors. This is too strong in

practice, since γ, γ' could require infinitely many bits of precision. However, we can perform Z_γ for $\gamma = \frac{C}{2^k}$ in quantum time and space of $O(k)$ (see for example [24]).

Combining these ideas, we can reformulate the (now almost) perfect amplitude amplification presented in [20]

Theorem 9 ((Almost) perfect quantum amplitude amplification, with known angle). *Let P be a projector acting on n qubits and O_P a projection oracle for P. Let $\mathcal{A}$ be a quantum unitary that produces a state $|\phi\rangle = \alpha |\phi_P\rangle + \beta \left|\phi_P^\perp\right\rangle$ where $|\phi_P\rangle \in Im(P)$ and $\left|\phi_P^\perp\right\rangle \in Ker(P)$. Suppose we know $|\alpha| = \sin(\theta)$ for some $\theta \in [0, \pi/2]$. There exists a quantum unitary circuit that:*

- *Consists of exclusively $N = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rfloor + 1$ calls to $O_P, O_P^\dagger, \mathcal{A}, \mathcal{A}^\dagger$ and a call to two different phase flips Z_γ and $Z_{\gamma'}$ where $\gamma, \gamma' \in [0, 1]$ are described with k bits of precision.*
- *Produces on input $|0\rangle$ a quantum state $|\phi_{output}\rangle$ satisfying*

$$|\langle \phi_{output} | \phi_P \rangle| = \sin(\beta) \quad with \left| \frac{\pi}{2} - \beta \right| \leq 2^{-k}.$$

and its running time is

$$\left(\left\lceil \frac{\pi}{4\theta} - \frac{1}{2} \right\rceil + 1 \right) (|\mathcal{A}|_{RT} + |O_P|_{RT}) + O(k).$$

Finally, we study what happens when we add some uncertainty to the angle θ. We are in the case where $|\theta - 2^{r/2}| \leq \frac{3}{2} \times 2^{-\frac{n}{2}} + o(2^{-n/2})$. Recall that depending on the problem we study, $r = \frac{2n}{5}$ or $r = \frac{2n}{7}$.

Proposition 2 (Almost perfect quantum amplitude amplification, with slightly unknown angle). *Let P be a projector acting on n qubits and O_P a projection oracle for P. Let $\mathcal{A}$ be a quantum unitary that produces a state $|\phi\rangle = \alpha |\phi_P\rangle + \beta \left|\phi_P^\perp\right\rangle$ where $|\phi_P\rangle \in Im(P)$ and $\left|\phi_P^\perp\right\rangle \in Ker(P)$. Suppose we know $|\alpha| = \sin(\theta)$ for some $\theta \in [0, \pi/2]$. There exists a quantum unitary circuit that:*

- *Consists of exclusively $N = \left\lfloor \frac{\pi 2^{r/2}}{4} - \frac{1}{2} \right\rfloor + 1$ calls to $O_P, O_P^\dagger, \mathcal{A}, \mathcal{A}^\dagger$ and a call to 2 different phase flips Z_γ and $Z_{\gamma'}$ where $\gamma, \gamma' \in [0, 1]$ are described with k bits of precision.*
- *Produces on input $|0\rangle$ a quantum state $|\phi_{output}\rangle$ satisfying*

$$|\langle \phi_{output} | \phi_P \rangle| = \sin\left(\pi/2 - 2^{-k} - 2N \times 2^{\frac{-n}{2}}\right) \approx \sin\left(\pi/2 - 2^{-k} - \frac{\pi}{2} 2^{\frac{r-n}{2}}\right).$$

and its running time is

$$\left(\left\lceil \frac{\pi}{4\theta} - \frac{1}{2} \right\rceil + 1 \right) (|\mathcal{A}|_{RT} + |O_P|_{RT}) + O(k).$$

here, we will usually take $k = \frac{n-r}{2}$ in order to have an error of $O\left(2^{\frac{r-n}{2}}\right)$ in the angle.

A.2 Uncertainty in the First QAA Algorithm

Our goal here is to create the state $|\phi_r\rangle = \sum_{x \in S_r} |x\rangle$ using $\texttt{QAA}(\texttt{setup}_{\mathcal{A}}, \texttt{proj}_{\mathcal{A}})$, where $\texttt{setup}_{\mathcal{A}}$ builds the superposition $|\phi_0\rangle = \frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle$ and $\texttt{proj}_{\mathcal{A}}$ projects onto the good subspace $span\{|x\rangle\}_{x \in S_r}$. We can write $|\phi_0\rangle = \alpha |\phi_r\rangle + \beta \left|\phi_r^{\perp}\right\rangle$ for some state $\left|\phi_r^{\perp}\right\rangle$ orthogonal to $|\phi_r\rangle$. Let also θ be such that $\alpha = \sin(\theta)$. We have $\alpha = \sqrt{\frac{|S_r^H|}{2^n}}$. Because H is a random function, there is some uncertainty in the value $|S_r^H|$, but it is in fact small.

Proposition 3. *With probability at least $\frac{99}{100}$ (over the randomness of the chosen function), we have*

$$|\theta - \theta_0| \leq 4 \times 2^{-\frac{n}{2}} + o(2^{-\frac{n}{2}}).$$

Where $\theta_0 = \arcsin(2^{-r/2})$.

Proof. As we said, the angle θ is closely related to the cardinal of the set S_r^H. We show the following lemma

Lemma 1. *With probability at least $\frac{99}{100}$ (for a random function H), we have $\left||S_r^H| - 2^{n-r}\right| \leq 4 \times 2^{\frac{n-r}{2}}$. We name this event "H is normal".*

Proof. Let $X_0, \ldots, X_{2^n-1}$ be 2^n independent variables such that $\forall i \in \{0,1\}^n, \Pr[X_i = 1] = 2^{-r}$ and $\Pr[X_i = 0] = 1 - 2^{-r}$, and let $X = \sum_{i \in \{0,1\}^n} X_i$. Let $\mathcal{F}_n$ be the set of random functions from $\{0,1\}^n$ to $\{0,1\}^n$. We have

$$\mathbb{E}_{H \leftarrow F_n}\left[\left|S_r^H\right|\right] = \mathbb{E}[X].$$

Using a Chernoff bound, we have

$$\Pr[|X - 2^{n-r}| \geq 4 \times 2^{\frac{n-r}{2}}] \leq 2e^{-\frac{16}{3}} \leq \frac{1}{100}$$

which allows us to conclude.

We use this lemma to bound θ. We know from our previous discussion that $\alpha = \sqrt{\frac{|S_r^H|}{2^n}}$ and recall that $\alpha = \sin(\theta)$. Using the previous lemma, we know that with probability at least $\frac{99}{100}$, we have $\left||S_r^H| - 2^{n-r}\right| \leq 4 \times 2^{\frac{n-r}{2}}$. This gives us $\left|\sin^2(\theta) - 2^{-r}\right| \leq 4 \times 2^{\frac{-n-r}{2}}$, which can be approximated, using $2^{\frac{-n+r}{2}} \ll 1$ as

$$|\sin(\theta) - 2^{-r/2}| \leq 2 \times 2^{-\frac{n}{2}}.$$

We introduce $\theta_0 = \arcsin(2^{-r/2})$ so we can write

$$|\theta - \theta_0| \leq 2 \times 2^{-\frac{n}{2}} + o(2^{-\frac{n}{2}}).$$

We can study our first QAA algorithm combining Proposition 2 and Proposition 3 to show the following.

Proposition 4. *In our algorithms for collision and multi-target preimage, each call to the first* `QAA` *algorithm runs in time*

$$\left(\left\lfloor \frac{\pi 2^{r/2}}{4} - \frac{1}{2} \right\rfloor + 1\right)(n + |O_H|_{RT}) + O(n-r).$$

where we count n the time to create a uniform superposition and $|O_H|_{RT}$ the running time of O_H. With probability $\frac{99}{100}$, all *those calls output a state $|\phi_{output}\rangle$ satisfying*

$$|\langle \phi_{output} | \phi_r \rangle \approx \cos((1+\pi)2^{\frac{r-n}{2}}).$$

An important thing to notice is that H is the same during the whole protocol so we have θ close to θ_0 as defined in Proposition 3, which happens with probability greater than $\frac{99}{100}$, for all runs.

A.3 Uncertainty in the General `QAA` Algorithm

In full generality, the inner `QAA` = (`setup`,`proj`) algorithm was supposed to produce the state $|\phi_r\rangle$ but produced instead the state $|\phi_{output}\rangle$. Notice that this setup is deterministic so it produces always the same $|\phi_{output}\rangle$.

In the case where the event "H is normal" occurs, we have

$$|\langle \phi_{output} | \phi_r \rangle \approx \cos((1+\pi)2^{\frac{r-n}{2}}).$$

Recall that $\texttt{proj} = \sum_{x: f_L^H(x)=1} |x\rangle\langle x|$. We have $tr(\texttt{proj}\,|\phi_r\rangle\langle\phi_r|) = \frac{|Sol_f|}{|S_r^H|}$. Both these quantities are random because H is random but we have a very good control over them. Using a similar Chernoff bound as before, we can show that with high probability, we have

- $|Sol_f| = 2 \times 2^{t-r} + O(2^{\frac{t-r}{2}})$.
- Since the event "H is normal" occurs, we know that $S_r^H = 2^{n-r} \pm 4 \times 2^{\frac{n-r}{2}}$.

From there, we clearly have that the ratio $tr(\texttt{proj}\,|\phi_r\rangle\langle\phi_r|) = \frac{|Sol_f|}{|S_r^H|} = 2^{t-n} + o(2^{t-n})$. Moreover,

$$|\langle \phi_{output} | \phi_r \rangle \approx \cos((1+\pi)2^{\frac{r-n}{2}}) = o(2^{\frac{t-n}{2}}).$$

Using the fact that the angle between pure states is a distance measure, we have $tr(\texttt{proj}\,|\phi_{output}\rangle\langle\phi_{output}|) = 2^{t-n} + o(2^{t-n})$.

The final `QAA` algorithm is applied $2^{\frac{n-t}{2}}$ times. Therefore, using Proposition 2 again, we can conclude that this protocol will succeed with probability $1 - o(1)$.