Quantum Linear Key-recovery Attacks Using the QFT

André Schrottenloher

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

Abstract. The Quantum Fourier Transform is a fundamental tool in quantum cryptanalysis. In symmetric cryptanalysis, hidden shift algorithms such as Simon's (FOCS 1994), which rely on the QFT, have been used to obtain structural attacks on some very specific block ciphers. The Fourier Transform is also used in classical cryptanalysis, for example in FFT-based linear key-recovery attacks introduced by Collard et al. (ICISC 2007). Whether such techniques can be adapted to the quantum setting has remained so far an open question.
In this paper, we introduce a new framework for quantum linear key-recovery attacks using the QFT. These attacks loosely follow the classical method of Collard et al., in that they rely on the fast computation of a *correlation state* in which experimental correlations, rather than being directly accessible, are encoded in the amplitudes of a quantum state. The experimental correlation is a statistic that is expected to be higher for the good key, and on some conditions, the increased amplitude creates a speedup with respect to an exhaustive search of the key. The same method also yields a new family of structural attacks, and new examples of quantum speedups beyond quadratic using classical known-plaintext queries.

Keywords: Linear cryptanalysis, Quantum cryptanalysis, Fast Walsh-Hadamard Transform, Quantum Fourier Transform

1 Introduction

Quantum cryptanalysis can be said to have started with Shor's algorithm [47], which showed that cryptosystems based on the hardness of factoring and computing discrete logarithms, which are secure classically, could be broken using a quantum computer. While Shor's algorithm provides an exponential speedup, at the other end of the spectrum, Grover's quantum search algorithm [25] provides a quadratic acceleration for NP problems, which is optimal for black-box search [4]. In particular, it halves the level of security for key-recovery provided by all ciphers.

Since then, many quantum algorithms have been designed and applied in cryptanalysis. In symmetric cryptanalysis, which is the main focus of this paper, they can be classified in two ways.

Q1 / Q2. Following a widely used terminology [34,33,29,28], Q1 adversaries are those which are capable of *offline* quantum computations, but only work from *classical* data. This is the most commonly used threat model in public-key post-quantum cryptography, underlying the ongoing standardization process organized by NIST [44]. In contrast, Q2 adversaries are capable of *quantum access* to oracles holding secret data (e.g., encryption, decryption, signing oracles). It is known since Kuwakado and Morii [38] that some symmetric cryptosystems are especially vulnerable to Q2 adversaries, while remaining secure against Q1 ones. For example, the Even-Mansour cipher is broken in Q2 [39] and secure in Q1 [2]. All Q2 breaks known to date rely on structure-finding algorithms: Simon's [48], Shor's, Kuperberg's [37], Bernstein-Vazirani [5], Deutsch-Josza [21].

Below quadratic / above quadratic. Starting from Grover's algorithm, one can build a family of *nested search* algorithms which reach at most a quadratic speedup. Most dedicated quantum attacks on symmetric cryptosystems so far belong to this family, with the notable exception of quantum slide attacks [33]. Notably, this category includes some Q2 attacks [34], collision attacks on hash functions [30] and a wide range of key-recovery techniques [24,20,12].

Better speedups than quadratic do not necessarily require Q2 queries, but all such attacks to date use the Quantum Fourier Transform in one way or another, usually a subcomponent of a structure-finding algorithm (Simon, Shor, *etc.*). The offline-Simon algorithm of Bonnetain et al. [11] was shown to reach a Q1 speedup of 2.5 for key-recovery on some block cipher constructions [13], i.e., from $\widetilde{\mathcal{O}}(2^{2.5n})$ to $\widetilde{\mathcal{O}}(2^n)$. Yamakawa and Zhandry achieved a more fundamental separation result [51]. They demonstrated that under a random oracle assumption, one can build a classically secure one-way function, which is quantumly invertible. That is, Q1 exponential speedups on various primitives (hash functions, block ciphers) are theoretically possible. However this separation has not been converted into an attack on practical constructions. Recently, Hosoyamada [28] achieved a (Q2) quantum speedup beyond quadratic on some types of integral distinguishers. His attack relies on a modified subroutine of Simon's algorithm and can be seen, like ours, as a *statistical* attack using the QFT.

Motivation and Contribution. The classical (fast) Fourier Transform is also a major tool of classical cryptanalysis. In particular, since the work of Collard et al. [16] it is used to speed up linear key-recovery attacks. It leads to the best attacks on well studied ciphers such as PRESENT [23], and several variants exist such as FFT-based zero correlation linear attacks [9]. However, while quantum linear attacks were investigated before [34,28], these works left as an open problem the use of the QFT in key-recovery attacks.

In this paper, we solve (partially) this long-standing open question, and introduce a new way to use the QFT in quantum key-recovery attacks. Our framework applies to the setting introduced by Collard et al. However, it comes with various limitations, and does not necessarily reach the same number of rounds as

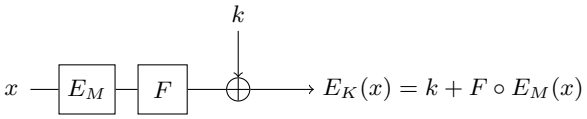

Fig. 1. Case considered by Collard et al. [16]. Under a guess of k, we have access to the middle rounds E_M (or a random permutation) and we compute the correlation of a linear approximation α, β of E_M.

classical attacks. But it can also do more: we show a 2.5 speedup on a structural attack, in a case where the offline-Simon algorithm [11,13] seems inapplicable.

Main Feature: The Correlation State. FFT-based linear cryptanalysis relies on the fast evaluation of *experimental correlations* $\widehat{\mathrm{cor}}(z)$ which depend on a subkey guess z and on a large database of N known-plaintext queries (see Figure 1). When guessing the good subkey right, we observe the reduced-round cipher E_M, which has a linear approximation, and the correlation is higher. One could evaluate each $\widehat{\mathrm{cor}}(z)$ in time N for a total $\mathcal{O}\big(N \times 2^{|k|}\big)$; however, Collard et al. showed how to evaluate all correlations in time $\widetilde{\mathcal{O}}\big(N + 2^{|k|}\big)$ using a fast Walsh-Hadamard transform.

Our main ingredient is a quantum analogue of this procedure which produces the *correlation state*:

$$|\mathsf{Cor}\rangle := \frac{1}{\sqrt{\sum_z \widehat{\mathrm{cor}}(z)^2}} \sum_z \widehat{\mathrm{cor}}(z)\,|z\rangle \quad .$$

Encoding correlations in the amplitude is also what Hosoyamada [28] did for quantum distinguishers. However, the quantum state that he constructed was a superposition of linear masks. We have a superposition of keys instead, since we are targeting key-recovery attacks. In this context, the construction of $|\mathsf{Cor}\rangle$ is more technical. It requires both the QFT and a *state preparation* technique, which is common in quantum algorithms. In fact, the principle is similar to [18], where the QFT is used to compute a *discrete convolution* of functions in the amplitudes of a quantum state.

From there, since the good subkey guess is expected to have a higher correlation, we use quantum amplitude amplification subroutines to complete the search for the key. This is the main limitation of our algorithm. Indeed, the speedup with respect to exhaustive search depends directly on the quality of the linear approximation (its *expected linear potential*, ELP). This can be seen in the statement of our main theorem, given in Section 4:

Theorem 1. *In the situation of Figure 1, let* $t = 1.005\sqrt{\mathrm{ELP}}2^{n/2}$*, where* ELP *is the ELP of the linear approximation* (α, β)*. There exists a quantum algorithm that takes no input and returns (after measurement) the master key with*

probability $\geq \frac{1}{2}$. *This algorithm has complexity:*

$$\frac{\pi^2}{8t} 2^{|K|/2} r(E) \mathrm{Tof}\,(E) + \frac{\pi^2}{8t} 2^{n/2} \mathrm{Tof}\,(\mathsf{CORCOMP}) \tag{1}$$

where $\mathrm{Tof}\,(E)$ *is the gate count of* E, $r(E)$ *the number of trial encryptions to test a key, and* $\mathrm{Tof}\,(\mathsf{CORCOMP})$ *is the gate count of computing the correlation state* $|\mathsf{Cor}\rangle$. *The attack succeeds with probability* 0.271.

New Structural Attack. When the experimental correlation for the right key becomes really large, the cipher E_M basically degenerates into a linear function, and our algorithm becomes a structural attack. We observe that this attack is different from the offline-Simon algorithm [11]. While offline-Simon typically requires chosen-plaintext queries, classical known-plaintext queries are sufficient in our case. Indeed, we use the QFT to compute a *statistic* (the correlation) instead of recovering a hidden structure.

Outline. In Section 2 we give preliminaries of linear cryptanalysis, notably the statistical models of experimental correlations of right and wrong key guesses, which are essential for our framework. In Section 3 we give some preliminaries of quantum algorithms. Our new algorithm is introduced in Section 4 and extended to multiple linear cryptanalysis in Section 5. Our applications are given in Section 6. We conclude the paper with several open questions in Section 7.

2 Preliminaries on Linear Cryptanalysis

In this section we give preliminaries on classical linear cryptanalysis, linear distinguishers and key-recovery attacks. We also recall *quantum* linear attacks which were proposed in [34]. From now on, we use $\cdot$ to denote the scalar product of vectors in $\mathbb{F}_2^n$ and $+$ for addition modulo 2 of bit-strings (including single bits).

2.1 Classical Linear Cryptanalysis

Linear cryptanalysis was introduced by Matsui [41] in order to attack the DES block cipher [42]. Let E_K be an n-bit block cipher instantiated with a given key K. A (keyless) *linear approximation* of E_K is a pair of n-bit *masks* $(\alpha, \beta) \in (\mathbb{F}_2^n)^2$. The *correlation* of this approximation is:

$$\mathrm{cor}_K(\alpha, \beta) := \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\alpha \cdot x + \beta \cdot E_K(x)} \quad . \tag{2}$$

Linear cryptanalysis exploits approximations with a large correlation. Matsui proposed two algorithms, Algorithm 1 and Algorithm 2, to perform a key-recovery. We will focus here on Algorithm 2, which targets a block cipher of the form $E_K = F_k \circ E'_K$, where E'_K has a correlated linear approximation and

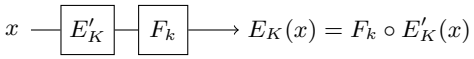

Fig. 2. Case considered in Matsui's Algorithm 2.

F_k is a keyed permutation (e.g., the last round) in which only a subset of the master key K, denoted k, intervenes. This is represented in Figure 2.

Algorithm 2 starts from a database D of N known plaintext-ciphertext pairs, and for each possible value z of k, it uses the database to compute the *experimental correlation*:

$$\widehat{\mathrm{cor}}(z) := \frac{1}{N} \sum_{(x,y)\in D} (-1)^{\alpha\cdot x+\beta\cdot F_z^{-1}(y)} \ . \tag{3}$$

Since it is a sum of N terms, this requires a total time $N \times 2^{|k|}$. The right key k is expected to be the one with the highest correlation (in absolute value). More generally, one keeps a certain proportion of subkeys having the largest correlations, and for each of these subkeys, one completes the key by exhaustive search.

ELP. As it can be seen in Equation 2, the correlation is a key-dependent quantity. The quality of an approximation (α, β) is measured over all the keys, using the *expected linear potential* (ELP):

$$\mathrm{ELP}(\alpha,\beta) := \frac{1}{2^{|K|}} \sum_{K\in\mathbb{F}_2^{|K|}} \mathrm{cor}_K(\alpha,\beta)^2 \ . \tag{4}$$

Statistical models for the experimental correlation were first formalized for single linear approximations [17,10] and then extended [7,8] for multiple linear approximations. These models depend on a factor B: $B = 1$ if the plaintexts are chosen uniformly at random with repetition and $B = \frac{2^n-N}{2^n-1}$ if they are distinct. In particular, if $N = 2^n$ (we know the whole codebook) we have $B = 0$. They also depend on whether the approximation has a *dominant trail*, or if it is a linear hull with many trails having a high correlation. In our applications, we focus on the second case. Here the experimental correlation for right and wrong keys follows normal distributions with different variances.

Assumption 1 (Right-key randomization hypothesis). The experimental correlation for the right subkey k ($\widehat{\mathrm{cor}}(k)$) is a random variable over k with normal distribution $\mathcal{N}(0, \sigma_R^2)$ where $\sigma_R^2 = \frac{B}{N} + \mathrm{ELP}$.

Assumption 2 (Wrong-key randomization hypothesis). Given a subkey k, the experimental correlation for wrong subkey guesses z ($\widehat{\mathrm{cor}}(z)$) is a random variable over z with normal distribution $\mathcal{N}(0, \sigma_W^2)$ where $\sigma_W^2 = \frac{B}{N} + 2^{-n}$.

If we want to keep only a proportion 2^{-a} of possible subkeys, we define a threshold $T = \sigma_W \Phi^{-1}(1 - 2^{-a-1})$, where Φ is the cumulative density function of

$\mathcal{N}(0,1)$. We will then keep all keys z with $|\widehat{\mathrm{cor}}(z)| \geq T$. This approach succeeds if the right key belongs indeed to this set, and the probability of this event is:

$$p := 2 - 2\Phi\left(\frac{\sigma_W}{\sigma_R}\Phi^{-1}(1-2^{-a-1})\right) \ .$$

Intuitively, for a constant a and p, we need σ_R to be bigger than σ_W by at least a constant factor, which gives that $N \times \mathrm{ELP}$ should be constant. So the data complexity of the attack is of order $N = \mathcal{O}\left(\mathrm{ELP}^{-1}\right)$.

2.2 Multiple Linear Cryptanalysis

Linear cryptanalysis becomes more powerful when we can use *multiple* linear approximations α_i, β_i [32]. These approximations do not need to relate to the same key and state bits [6].

Consider M approximations. The correlation is replaced by the *capacity*:

$$C(K) = \sum_{i=1}^{M} \mathrm{cor}_K(\alpha_i, \beta_i)^2 \ . \tag{5}$$

The capacity is estimated by summing the correlations for representative families of trails for each approximation. If we can include all M approximations in this computation, we obtain an estimate C such that: $\mathrm{Exp}_K(C(K)) \simeq C + M2^{-n}$ and $\mathrm{Var}_K(C(K)) \simeq \frac{2}{M}C^2$. (Theorem 4.5 in [7]). The corresponding experimental statistic is:

$$\widehat{q}(z) = \sum_{i=1}^{M} \widehat{\mathrm{cor}}_i(z)^2 \ . \tag{6}$$

We use the results of [7] (Theorem 6) for the distributions of the statistics of the right and wrong key, which hold under an assumption of independence of the approximations.

Assumption 3 (Right-key randomization hypothesis, multiple). The statistic $\widehat{q}(k)$ for the right subkey k is a random variable over K following a normal distribution $\mathcal{N}(\mu_R, \sigma_R^2)$ where

$$\begin{cases} \mu_R = \frac{B}{N}M + \mathrm{Exp}_K(C(K)) \\ \sigma_R^2 = 2\frac{B^2}{N^2}M + 4\frac{B}{N}\mathrm{Exp}_K(C(K)) + \mathrm{Var}_K(C(K)) \ . \end{cases} \tag{7}$$

Assumption 4 (Wrong-key randomization hypothesis, multiple). Given a subkey k, the statistic $\widehat{q}(z)$ for wrong subkey guesses z follows a multiple of a χ^2 distribution with M degrees of freedom: $\frac{B+N2^{-n}}{N}\chi^2_M$, so with average and variance:

$$\begin{cases} \mu_W = M\left(\frac{B}{N} + 2^{-n}\right) \\ \sigma_W^2 = 2M\left(\frac{B}{N} + 2^{-n}\right)^2 \ . \end{cases} \tag{8}$$

With $B = 0$ and $N = 2^n$, these parameters are simplified into:

$$\begin{cases} \mu_R = \mathrm{Exp}_K(C(K)) \simeq C + M2^{-n}, \sigma_R^2 = \mathrm{Var}_K(C(K)) \simeq \frac{2}{M}C^2 \\ \mu_W = M2^{-n}, \sigma_W^2 = 2M2^{-2n} \ . \end{cases} \tag{9}$$

2.3 Advanced key-recovery Attacks and the FFT

Once the linear approximation has been chosen, we must find the best strategy to evaluate the experimental correlations $\widehat{\mathrm{cor}}(z)$ for all z and filter out the z exceeding a selected threshold. One can usually do better than the generic $N \times 2^{|k|}$ using an early-abort technique. We guess only the necessary key bits to compute a sequence of intermediate tables, which count the number of plaintext-ciphertext pairs leading to certain internal state values. An example of this technique can be found in [49]. However, on many ciphers like PRESENT [23] or SIMON [40], the best key-recovery attacks use the FFT approach introduced by Collard et al. [16].

We focus on the situation studied in [16], represented in Figure 1, which is closer to our situation in the quantum setting. We note that it was extended afterwards in [23] with key additions in both the first and last rounds.

Hadamard Transform. The *Walsh-Hadamard Transform* (WHT) is a special case of the discrete Fourier transform. Given a function $f : \mathbb{F}_2^n \to \mathbb{C}$, its transform $\hat{f}$ is defined as:

$$\begin{cases} \hat{f} : \mathbb{F}_2^n \to \mathbb{C} \\ x \mapsto \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y} f(y) \end{cases} \tag{10}$$

Note that by convention, we do not normalize it (contrary to the quantum Hadamard transform). The Fast Walsh-Hadamard Transform (FWHT) algorithm is a special case of FFT which evaluates the WHT of f in time $\mathcal{O}(n2^n)$.

Considering the database D of N known-plaintext queries, we let "$(x, *) \in D$" be the predicate that determines if x is among the plaintexts, $(*, x) \in D$ to determine if x is among the ciphertexts, and for a given x, we let $D(x)$ and $D^{-1}(x)$ be the corresponding ciphertext (resp. plaintext).

Recall that for each $z \in \mathbb{F}_2^n$, we need to evaluate the experimental correlation:

$$\begin{aligned} \widehat{\mathrm{cor}}(z) &= \frac{1}{N} \sum_{(x,*) \in D} (-1)^{\alpha \cdot x + \beta \cdot F^{-1}(z + D(x))} \\ &= \frac{1}{N} \sum_{y \in \mathbb{F}_2^n} \mathbf{1}\left[(*, y) \in D\right] (-1)^{\alpha \cdot D^{-1}(y)} (-1)^{\beta \cdot F^{-1}(z+y)} \ . \end{aligned}$$

Adapting [16], we introduce the pair of functions f, g:

$$\forall x \in \mathbb{F}_2^n \begin{cases} f(x) := \mathbf{1}\left[(*, x) \in D\right] (-1)^{\alpha \cdot D^{-1}(x)} \\ g(x) := (-1)^{\beta \cdot F^{-1}(x)} \end{cases} \tag{11}$$

The experimental correlation is actually the *discrete convolution* of f and g at z:

$$\widehat{\mathrm{cor}}(z) = \frac{1}{N} \sum_{y \in \mathbb{F}_2^n} f(y) g(y + z) := \frac{1}{N} \left(f \star g\right)(z) \ . \tag{12}$$

In order to evaluate $(f \star g)$, we use the *convolution theorem*: the convolution of two functions is equal, under a Fourier Transform, to the pointwise product of their Fourier Transforms. In our case:

$$(f \star g) = \frac{1}{2^n} \widehat{\widehat{f} \cdot \widehat{g}} \ . \tag{13}$$

The complexity to compute all correlations is thus reduced from $N2^{|k|}$ to $\mathcal{O}\big(n(N + 2^{|k|})\big)$, since we only need to do WHTs and pointwise products of vectors of length $2^{|k|}$.

2.4 Quantum Linear Cryptanalysis

In [34, Section 7], Kaplan et al. showed that quantum search (Grover search and Amplitude Amplification) could be used to speedup some classical linear key-recovery attacks. The proposed attack is a last-rounds attack similar to Matsui's algorithm 2, using either Q1 or Q2 queries.

With Q1 queries, it has a complexity $\mathcal{O}\Big(N + 2^{|k|/2}\sqrt{N}\Big)$ (and then the partial key must be completed). Note that this assumes that the good subkey k can be identified by its correlation, and that there are no false positives. While the first step (obtaining the data) is not accelerated, the second uses a Grover search on the possible values of k, and approximate counting to estimate the correlation for a given k (in time $\sqrt{N}$).

If quantum queries are given, then the data collection step is not required anymore, and the complexity becomes $\mathcal{O}\Big(2^{|k|/2}\sqrt{N}\Big)$. Also, this method uses a single linear approximation. In the case of *multiple* linear cryptanalysis, it may work only if we can guess globally the $|k|$ bits of key required for *all* linear approximations at the same time, which is rarely the case in advanced attacks.

Thus, an important characterization of these known attacks is that they cannot reach more rounds than the classical attacks that use a *single approximation and no FFT*.

New Approach for Distinguishers. Recently Hosoyamada [28] used a procedure inspired by Simon's algorithm to speedup some linear distinguishers. The main idea is that, using a single Q2 query to a function $f : \{0,1\}^n \to \{0,1\}^n$, one can produce a superposition:

$$\frac{1}{\sqrt{2^n}} \sum_{(\alpha,\beta)\in(\mathbb{F}_2^n)^2} \sum_x (-1)^{\alpha\cdot x+\beta\cdot f(x)} \left|\alpha\right\rangle \left|\beta\right\rangle \ .$$

It can be seen that the correlations appear directly in the amplitudes. Thus, to distinguish a function f having a correlated approximation (α, β), it suffices to estimate the corresponding amplitude, with typically a quadratic speedup with respect to classical estimates of the correlation.

The important difference with [34] is that this approach also speeds up multidimensional linear distinguishers. Besides, for some *multiple multi-dimensional*

integral distinguishers, the amplitude on the component α, β can become so large that the speedup is better than quadratic. However, extending this to key-recovery attacks remains an open question.

The similarity of this approach with our work is evident, as in both cases, one obtains a quantum state with amplitudes encoding a *statistic* (the correlation). The difference is that we have a superposition of *keys*, instead of a superposition of masks. The subroutine that computes our own correlation state is also much different from the one in [28]. In Hosoyamada's attacks, the statistic appears with a singe Hadamard transform, while our procedure involves a discrete convolution, which will be detailed in Section 4.

3 Preliminaries of Quantum Computing

In this section, we collect some important preliminaries of quantum computing and detail the important building block of *state preparation*.

3.1 Quantum Computing Basics

We assume some basic knowledge of the quantum circuit model, e.g., quantum gates, operators, measurements [43]. In this paper, we will measure the time complexity of a quantum circuit as its *Toffoli* gate count, written Tof (). Indeed, all the circuits that we consider below are entirely made of Clifford and Toffoli gates; Clifford gates are often considered cheaper.

qRAM and Q2 queries. A quantum algorithm can make use of different types of memory: classical memory, but also *quantum-accessible* and *quantum* memory, which is often denoted as qRAM. In this paper we consider quantum-accessible classical memory (QRACM). This is a special hardware holding classical bits, but allowing *quantum access*, i.e., an efficient implementation of the following unitary:

$$|b\rangle |i\rangle \xmapsto{\mathsf{qRAM}(y_1,\ldots,y_M)} |b + y_i\rangle |i\rangle \tag{14}$$

where M is the number of bits of the QRACM and $y_1, \ldots, y_M$ its contents. Notice that such a circuit could be implemented using about $\mathcal{O}(M)$ standard gates. In the *QRACM model*, we allow such access in time $\mathcal{O}(1)$.

When analyzing a block cipher E_K in the Q2 model, we assume that superposition queries are available via a unitary: $|x\rangle |0\rangle \mapsto |x\rangle |E_K(x)\rangle$. Such an operator can also be realized by storing classical queries in QRACM, which is why our attacks can use either QRACM, Q2 queries, or both. From this perspective, we consider a QRACM query to cost as much as a block cipher query.

Elementary Arithmetic Operations. We recall gate counts for some standard arithmetic operations, which can be found in the literature.

– Addition: adding two integers modulo 2^n can be done with $2n - 1$ Toffoli gates [50]. A *controlled* addition circuit can be done with $3n+3$ Toffoli gates.

– Comparison: comparing two n-bit numbers can be done with the same cost as an addition or subtraction (e.g., we can subtract the numbers and observe the sign of the result), so we consider a cost of $2n-1$ Toffoli gates as well.
– Multiplication: multiplying two integers modulo 2^n can be done with a sequence of n controlled additions and shifts (see e.g. Appendix A and B in [27]). We evaluate the corresponding number of Toffoli gates to $\sum_{i=1}^{n} 3(n-i) + 3n = \frac{3}{2}n^2 + \frac{3}{2}n$.

3.2 Quantum Search

Quantum Amplitude Amplification [14] (QAA) is a generalization of Grover's algorithm [25] which amplifies the probability of success of any quantum algorithm. Let $\mathcal{A}$ be a quantum algorithm that produces a superposition of a "good state" $|\psi_G\rangle$ and a "bad state" $|\psi_B\rangle$ of the form:

$$\mathcal{A}|0\rangle = \sqrt{p}\,|\psi_G\rangle\,|1\rangle + \sqrt{1-p^2}\,|\psi_B\rangle\,|0\rangle \tag{15}$$

where p is the probability of success of $\mathcal{A}$. Let O_{test} be an operator which flips the phase in the case 1 only. Let O_0 be an operator called *inversion around zero*, that flips the phase of the basis vector $|0\rangle$ (and only this one). The QAA starts from the output of $\mathcal{A}$: $|\psi_0\rangle := \mathcal{A}|0\rangle$ and constructs a sequence of states $|\psi_{i+1}\rangle := -\mathcal{A}O_0\mathcal{A}^\dagger O_{\text{test}}|\psi_i\rangle$. Its main property is:

Lemma 1 (From [14]). *Let θ be such that $\theta = \arcsin\sqrt{p}$. Then: $|\psi_i\rangle = \sin((2i+1)\theta)\,|\psi_G\rangle\,|1\rangle + \cos((2i+1)\theta)\,|\psi_B\rangle\,|0\rangle$.*

This is shown with a geometric argument: the QAA operator $\mathcal{A}O_0\mathcal{A}^\dagger O_{\text{test}}$ realizes a rotation of angle 2θ in the plane spanned by $|\psi_G\rangle\,|1\rangle$ and $|\psi_B\rangle\,|0\rangle$.

Exhaustive Key Search. For a block cipher E_K, exhaustive key search using Grover's algorithm consists in finding among all possible keys the one that matches a few known plaintext-ciphertext pairs. This requires $\frac{\pi}{2}2^{|K|/2}r(E)\text{Tof}\,(E)$ Toffoli gates, where $r(E)$ is the number of trial encryptions required to discriminate the good key with certainty. In fact, the factor $r(E)$ can be amortized to 1 [19], both in exhaustive search and in our attacks. However, we will keep it to simplify the analysis.

Exact QAA and Unknown Success Probability. When the success probability is known exactly, it is possible to construct the state $|\psi_G\rangle$ exactly using a final *partial* rotation that reaches an angle $\frac{\pi}{2}$ [14]. The total number of iterates is thus $\left\lceil \frac{\pi}{4\arcsin\sqrt{p}} \right\rceil \leq \frac{\pi}{4\sqrt{p}} + 1$. The implementation of this final operator is handled via the Solovay-Kitaev theorem (see e.g. [46] for efficient implementations of arbitrary rotation operators). As it will not dominate the complexity anyway, we will not enter its details here.

When an interval on the success probability is known, performing an Exact QAA is still a good strategy: the relative error on p does not increase after amplification.

Algorithm 1 Main subroutine of quantum state preparation (from [45], adapted to handle negative values).

Input: Q2 access to E_K
Output: returns the master key K or fails

1: Call $\mathcal{A}$ ▷ $\sum_x \alpha_x |x\rangle$
2: Call $\mathcal{B}$ ▷ $\sum_x \alpha_x |x\rangle |f(x)\rangle$
3: Flip the phase depending on the sign of $f(x)$ ▷ $\sum_x \alpha_x \mathrm{sgn}(f(x)) |x\rangle |f(x)\rangle$
4: Create a uniform superposition over $[0; 2^n - 1]$ in a new register, using Hadamard gates
▷ $\sum_x \alpha_x \mathrm{sgn}(f(x)) |x\rangle |f(x)\rangle \frac{1}{2^{n/2}} \sum_{0 \leq y \leq 2^n - 1} |y\rangle$
5: Compare the value of y with $|f(x)|$ and write the result in a new output qubit
6: Apply $\mathcal{B}$ to uncompute $f(x)$
▷ $\sum_x \alpha_x \mathrm{sgn}(f(x)) |x\rangle \frac{1}{2^{n/2}} \left(\sum_{0 \leq y \leq |f(x)| - 1} |y\rangle |0\rangle + \sum_{|f(x)| \leq y \leq 2^n - 1} |y\rangle |1\rangle \right)$
7: Apply Hadamard gates on the register holding y

Lemma 2 (Lemma 5 in [12]). *Assume that $\mathcal{A}$ has a success probability $p' \in [p(1-\varepsilon); p(1+\varepsilon)]$ for $\varepsilon \leq \frac{1}{2}$. After running an exact QAA that assumes a success probability exactly equal to p, the success probability becomes greater than $1 - \varepsilon^2$.*

3.3 State Preparation for Amplitude Products

One of the main ideas of our algorithm is to perform computations *in the amplitude*. In particular, we need to multiply the amplitudes of a quantum state by values given separately by an oracle.

Let X be a set (identified with a set of bit-strings). Let $\mathcal{A}$ and $\mathcal{B}$ be two unitary operators (quantum circuits without measurements) such that: $\mathcal{A} |0\rangle = \sum_{x \in X} \alpha_x |x\rangle$ and $\mathcal{B} |x\rangle = |x\rangle |f(x)\rangle$, where $f : X \to]\!] -2^n; 2^n [\![$. In other words, we have a quantum circuit that produces a superposition and another that computes an integer function. Our goal is to multiply the amplitudes by $f(x)$ and renormalize, that is, obtain the state:

$$\frac{1}{\sqrt{\sum_{x \in X} \alpha_x^2 f(x)^2}} \sum_{x \in X} \alpha_x f(x) |x\rangle \quad .$$

This is a generalization of *state preparation*, where we would have $\alpha_x = \frac{1}{\sqrt{|X|}}$. A generic method for black-box state preparation was given by Grover [26], but it relies on heavy quantum arithmetic circuits. In this paper, we use the more lightweight method of Sanders *et al.* [45]. The main subroutine is given in Algorithm 1.

Lemma 3. *There exists a unitary U such that:*

$$U |0\rangle = \sum_x \alpha_x \frac{f(x)}{2^n} |x\rangle |0\rangle + |\phi\rangle$$

where $|\phi\rangle$ is a non-normalized state where the last qubits are not zero. The Toffoli gate count of U is upper bounded by:

$$\mathrm{Tof}\,(U) \leq 2\mathrm{Tof}\,(\mathcal{B}) + \mathrm{Tof}\,(\mathcal{A}) + 2n - 1 \ .$$

Proof. After running Algorithm 1, we obtain a state of the form:

$$\sum_x \alpha_x \mathrm{sgn}(f(x)) \frac{|f(x)|}{2^n} |x\rangle |0\rangle |0\rangle + |\phi\rangle \tag{16}$$

where $|\phi\rangle$ is a non-normalized state with either the last flag equal to 1, or the y register different from zero. □

Afterwards, we must amplify the part of the state with flag zero, in order to obtain the exact superposition that we want.

4 Our new Algorithm

Recall the situation presented in Section 2.3: the experimental correlations for key guesses can be expressed as the convolution of two functions. In the quantum setting, with a similar sequence of operations, we will show how to compute the convolution into the amplitudes:

$$\frac{1}{\sqrt{\sum_z (f \star g)(z)^2}} \sum_z (f \star g)(z) |z\rangle$$

We know that the right key guess will have a higher amplitude in this superposition. However, as it remains rather small, trying to measure directly this state would be useless. Instead, we use this as the starting point of another QAA which tries to complete the whole key.

4.1 Situation

We now consider the generic situation depicted in Figure 3, which is a hybrid between [16] and [23]. We define 3 subsets of key bits: • k^{in} are the *inner* key bits (they need to be guessed first); • k^{out} are the *outer* key bits (they are handled by the WHT); • k^c are the key bits which allow to complete the master key.

We assume that there are s bit relations between k^{in} and k^{out}, and that a choice of agreeing $k^{\mathrm{in}}, k^{\mathrm{out}}, k^c$ determines completely the master key. We also assume that $|k^{\mathrm{out}}| = n$, i.e., like in Collard et al.'s initial attack, the FWHT is computed on the whole state size. With all these definitions, we have: $|K| = (|k^{\mathrm{in}}| + |k^{\mathrm{out}}| - s) + |k^c|$.

We assume that E_M admits a linear approximation (α, β) without a dominant trail, with a certain ELP. We assume that a database D of N known plaintext queries $(x, E_K(x))$ is given. We use the abbreviation "$x \in D$" to denote that the

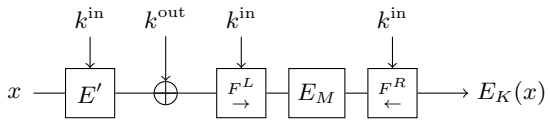

Fig. 3. Our generic attack. While E' is a permutation, both F_L (computed forwards) and F_R (computed backwards) are functions which determine the value in the input and output masks.

plaintext x belongs to the database and $D(x) := E_K(x)$ in that case. For each guess of inner key z^{in} and outer key z^{out}, the experimental correlation is:

$$\begin{aligned}\widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}}) &= \frac{1}{N}\sum_{x\in D}(-1)^{\beta\cdot F^R_{z^{\text{in}}}(D(x))}(-1)^{\alpha\cdot F^L_{z^{\text{in}}}(z^{\text{out}}+E'_{z^{\text{in}}}(x))} \\ &= \frac{1}{N}\sum_{x\in\mathbb{F}_2^n}\mathbf{1}\left[x\in D\right](-1)^{\beta\cdot F^R_{z^{\text{in}}}(D(x))}(-1)^{\alpha\cdot F^L_{z^{\text{in}}}(z^{\text{out}}+E'_{z^{\text{in}}}(x))} \\ &= \frac{1}{N}\sum_{x\in\mathbb{F}_2^n}\mathbf{1}\left[E'^{-1}_{z^{\text{in}}}(x)\in D\right](-1)^{\beta\cdot F^R_{z^{\text{in}}}\circ D\circ E'^{-1}_{z^{\text{in}}}(x))}(-1)^{\alpha\cdot F^L_{z^{\text{in}}}(z^{\text{out}}+x)} \quad .\end{aligned}$$

Thus, even in the case of reduced data, we can still define the experimental correlations as the convolution of two functions. For each z^{in}, we define:

$$\forall x\in\mathbb{F}_2^n, \begin{cases} f_{z^{\text{in}}}(x) = \mathbf{1}\left[E'^{-1}_{z^{\text{in}}}(x)\in D\right](-1)^{\beta\cdot F^R_{z^{\text{in}}}\circ D\circ E'^{-1}_{z^{\text{in}}}(x))} \\ g_{z^{\text{in}}}(x) = (-1)^{\alpha\cdot F^L_{z^{\text{in}}}(z^{\text{out}}+x)} \\ \widehat{\text{cor}}(z^{\text{in}}, z) = \frac{1}{2^n}\left(f_{z^{\text{in}}}\star g_{z^{\text{in}}}\right)(z) \\ |\mathsf{Cor}_{z^{\text{in}}}\rangle = \frac{1}{\sqrt{\sum_z \widehat{\text{cor}}(z^{\text{in}},z)^2}}\sum_z \widehat{\text{cor}}(z^{\text{in}}, z)\,|z\rangle \quad . \end{cases} \tag{17}$$

By assumption, in the state $|\mathsf{Cor}_{k^{\text{in}}}\rangle$, there is a bigger amplitude on the basis state k^{out}. This is what we want to exploit; we start with the construction of $|\mathsf{Cor}_{z^{\text{in}}}\rangle$, and around this, we build several layers of QAA to complete the search for the good key.

4.2 Construction and Analysis of $|\mathsf{Cor}\rangle$

First, we must make some assumptions. We assume that efficient unitaries are given for F_R, F_L and E', of gate counts Tof (F_R), Tof (F_L) and Tof (E'). In order to compute $f_{z^{\text{in}}}$, we need either: • Q2 queries (in that case, $N = 2^n$); • a large QRACM storing the database D. In both cases, we have access to two unitaries:

$$\begin{cases} \mathsf{INIT} & : |0\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{x\in D}|x\rangle \\ \mathsf{QUERY} & : |x\rangle\,|0\rangle \mapsto |x\rangle\,|D(x)\rangle \end{cases} \tag{18}$$

where INIT can be implemented with an appropriate data structure, and QUERY is either a Q2 query, or a QRACM query which is undefined if $x \notin D$.

Algorithm 2 Subroutine of CORCOMP (Lemma 4). It runs in two phases: 1. Computation of f in the amplitude (Steps 1 to 7), 2. Fourier transform and multiplication by $\widehat{g}$ (Steps 8 to 9).

Input: state $|z^{\text{in}}\rangle\,|0_n\rangle$

1: Initialize ancilla registers ▷ $|z^{\text{in}}\rangle\,|0_n\rangle\,|0_n\rangle\,|0_n\rangle$

2: Apply INIT ▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_{x\in D}|x\rangle\,|0_n\rangle\,|0_n\rangle$

3: Apply E' in place (needs to compute E' and E'^{-1})

▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_{x\in D}|E'_{z^{\text{in}}}(x)\rangle\,|0_n\rangle\,|0_n\rangle = |z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_{x\in\mathbb{F}_2^n}\mathbf{1}\left[E'^{-1}_{z^{\text{in}}}(x)\in D\right]|x\rangle\,|0_n\rangle\,|0_n\rangle$

4: Apply QUERY

▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_x \mathbf{1}\left[E'^{-1}_{z^{\text{in}}}(x)\in D\right]|x\rangle\,|E_K(x)\rangle\,|0_n\rangle$

5: Apply F^R ▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_x \mathbf{1}\left[E'^{-1}_{z^{\text{in}}}(x)\in D\right]|x\rangle\,|E_K(x)\rangle\,|F^R\circ E_K(x)\rangle$

6: Compute the dot-product with α in the phase

▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_x f_{z^{\text{in}}}(x)\,|x\rangle\,|E_K(x)\rangle\,|F^R\circ E_K(x)\rangle$

7: Re-apply F^R and QUERY ▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N}}\sum_x f_{z^{\text{in}}}(x)\,|x\rangle\,|0_n\rangle\,|0_n\rangle$

8: Apply H ▷ $|z^{\text{in}}\rangle\,\frac{1}{\sqrt{N2^n}}\sum_x \widehat{f_{z^{\text{in}}}}(x)\,|x\rangle\,|0_n\rangle\,|0_n\rangle$

▷ Forget about the ancilla registers

9: Apply the state preparation technique of Lemma 3

▷ $|z^{\text{in}}\rangle\left(|\phi\rangle + \frac{1}{G\sqrt{N2^n}}\sum_y(\widehat{f_{z^{\text{in}}}}\widehat{g_{z^{\text{in}}}})(y)\,|y\rangle\,|0\rangle\right)$

▷ Here $|\phi\rangle$ is a non-normalized vector whose details are insignificant for the rest of the algorithm

Finally, we need a unitary that computes the Fourier coefficients of $g_{z^{\text{in}}}$. In some cases this may be done with a precomputation, but otherwise, we will have to implement this unitary "by hand":

$$\text{GFOURIER} : |z^{\text{in}}\rangle\,|x\rangle\,|0\rangle \mapsto |z^{\text{in}}\rangle\,|x\rangle\,|\widehat{g_{z^{\text{in}}}}(x)\rangle \quad . \tag{19}$$

For $z^{\text{in}} = k^{\text{in}}$, let G be an upper bound on the absolute value of Fourier coefficients of $g_{k^{\text{in}}}$. For a random permutation, G is of order $\mathcal{O}\left(\sqrt{n}2^{n/2}\right)$, as we show in Appendix A.

Lemma 4. *There exists an algorithm* CORCOMP *returning a state of the form:*

$$\text{CORCOMP}\,|z^{\text{in}}\rangle\,|0\rangle = |z^{\text{in}}\rangle\,|\text{Cor}_{z^{\text{in}}}\rangle \quad .$$

The Toffoli gate count of CORCOMP *is given by:*

$$\begin{aligned}\text{Tof}\,(\text{CORCOMP}) \leq \left(\frac{\pi}{2}\frac{G}{2^{n/2}}+3\right)\Big[&\text{Tof}\,(\text{INIT})+2\text{Tof}\,(\text{QUERY})\\ &+2\text{Tof}\,(\text{GFOURIER})+\text{Tof}\,(E')+\text{Tof}\left(E'^{-1}\right)\\ &+2\text{Tof}\left(F^R\right)+\left(2\left\lceil\log_2 G\right\rceil-1\right)+2\left(n+\left\lceil\log_2 G\right\rceil\right)\Big] \quad .\end{aligned} \tag{20}$$

Proof. We start from Algorithm 2. In the superposition at the end, the probability to measure the flag 1 is equal to:

$$p := \frac{1}{G^2 2^{2n}} \sum_z (\widehat{f_{z^{\text{in}}}} \widehat{g_{z^{\text{in}}}})^2 = \frac{1}{G^2 2^n} \sum_z \left(f_{z^{\text{in}}} \star g_{z^{\text{in}}}\right)(z)^2 = \frac{2^n}{G^2} \sum_z \widehat{\text{cor}}(z^{\text{in}}, z)^2 \ .$$

To estimate $\sum_z \widehat{\text{cor}}(z^{\text{in}}, z)^2$, we neglect the case of the good key $z = k^{\text{out}}$, since its contribution to the sum will remain negligible. We use only the wrong-key randomization hypothesis, assuming that the $\widehat{\text{cor}}(z^{\text{in}}, z)$ for wrong z^{in} and z follow a normal distribution $\mathcal{N}(0, 2^{-n})$ (since we have the full codebook). So $2^{n/2}\widehat{\text{cor}}(z^{\text{in}}, z)$ follows a normal distribution $\mathcal{N}(0,1)$ and $2^n \sum_z \widehat{\text{cor}}(z^{\text{in}}, z)^2$ follows a $\chi^2_{2^n}$ distribution with mean 2^n and variance 2^{n+1}. Using Chebyshev's inequality:

$$\Pr\left(\left|2^n \sum_z \widehat{\text{cor}}(z^{\text{in}}, z)^2 - 2^n\right| \geq 10 \cdot 2^{(n+1)/2}\right) \leq \frac{1}{100} \ . \tag{21}$$

Thus, with 99% chance over the run of the attack, we have the bound on p:

$$\left|p - \frac{2^n}{G^2}\right| \leq \frac{10 \cdot 2^{(n+1)/2}}{G^2} \ . \tag{22}$$

The Toffoli gate count is given by putting together the different operations.

In order to eliminate the component $|0\rangle$, we apply an Exact QAA over Algorithm 2, assuming that the success probability is exactly $\frac{2^n}{G^2}$. The term $2(n + \lceil \log_2 G \rceil)$ is due to the inversion around zero in the QAA. The minor relative error on the success probability does not disrupt QAA computations (see Lemma 2).

Finally, we apply a final Hadamard transform to obtain the correlations. □

From now on, unless stated otherwise, we will consider $N = 2^n$ (full codebook available), as it simplifies the computations and makes INIT trivial. Besides, we will always neglect the terms $T(E') + T(E'^{-1}) + 2T(F^R) + (2\lceil \log_2 G \rceil - 1) + 2(n + \lceil \log_2 G \rceil)$, as we expect all of them to cost much less than complete block cipher evaluations. The cost of each iterate becomes dominated by 2Tof (QUERY) + 2Tof ($\mathsf{GFOURIER}$).

Analysis of $|\mathsf{Cor}_{k^{\text{in}}}\rangle$. Our attack works because the amplitude on the right key in $|\mathsf{Cor}_{k^{\text{in}}}\rangle$ is bigger. To quantify how much, we need to bound $|\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})|$. We write both an upper and a lower bound using some threshold t, as:

$$t 2^{-n/2} \leq |\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})| \leq 2t 2^{-n/2} \ . \tag{23}$$

For fixed t, since $\frac{|\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})|}{\sqrt{\text{ELP}}}$ follows a normal distribution $\mathcal{N}(0,1)$, the probability that this event happens (over the run of the attack) is equal to:

$$2\left(\Phi\left(\frac{2t2^{-n/2}}{\sqrt{\text{ELP}}}\right) - \Phi\left(\frac{t2^{-n/2}}{\sqrt{\text{ELP}}}\right)\right) \ .$$

Algorithm 3 QAA architecture for the key-recovery, using CORCOMP as building block, in the situation represented on Figure 3.

Input: access to E_K (Q2 or QRACM)
Output: returns the master key K or fails

1: **Run QAA over the following:**
2: Create a uniform superposition of z^{in}
3: **Run QAA over the following:**
4: Run CORCOMP
5: Test if z^{in} agree with z^{out}
6: **EndQAA**
7: **Run QAA over the following:**
8: Create a uniform superposition of z^c
9: Write 1 if $z^{\text{in}}, z^{\text{out}}, z^c$ leads to the correct K
10: **EndQAA**
11: Check if the flag is 1
12: **EndQAA**

CORFILT (Lemma 5)

SETUP (Lemma 6)

Later on, we will see that the complexity of the attack is proportional to $\frac{1}{t}$. In order to minimize it for a given ELP, we must choose $c = \frac{t2^{-n/2}}{\sqrt{\text{ELP}}}$ such that $2c(\Phi(2c) - \Phi(c))$ is maximal. Via numerical optimization we find that $c = 1.005$ gives the maximal value 0.272. For this value of c we also have $2(\Phi(2c) - \Phi(c)) = 0.271$. Thus by selecting $t = 1.005\sqrt{\text{ELP}}2^{n/2}$ we are ensured that Equation 23 holds with probability at least 0.271.

4.3 QAA Layers

Starting from the computation of $|\mathsf{Cor}_{z^{\text{in}}}\rangle$, several layers of QAA are necessary to complete our algorithm, given in Algorithm 3. We analyze each level carefully in a bottom-up approach, starting from the innermost level and computing the probability of success of each QAA.

As we have seen, CORCOMP outputs a superposition of outer keys; however it does not take into account the relations with the inner key. Classically, these relations reduce the number of degrees of freedom and can be used in conjunction with a pruned Walsh transform [23]. However, in our case they are problematic. To eliminate them, we need to perform another layer of QAA.

Lemma 5. *There exists an algorithm* CORFILT *such that:*

$$\text{CORFILT}\,|z^{\text{in}}\rangle\,|0\rangle = |z^{\text{in}}\rangle\,2^{s/2} \sum_{z^{\text{in}}, z^{\text{out}}\ agree} \widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}})\,|z^{\text{out}}\rangle \quad . \tag{24}$$

Its gate count is given by:

$$\text{Tof}\,(\text{CORFILT}) = \left(\frac{\pi}{2}2^{s/2} + 3\right)\text{Tof}\,(\text{CORCOMP}) \tag{25}$$

Proof. Note that we have simplified the writing by approximating $\sum_z \widehat{\mathrm{cor}}(z)^2 \simeq 1$.

The structure of CORFILT is simply an Exact QAA performed over the output of CORCOMP, to force that z^{out} agrees with z^{in}. The number of iterates depends on the probability, on the output of CORCOMP, that z^{in} and z^{out} agree. For a given z^{in}, and setting aside the right key case, this is a sum of 2^{n-s} squared random correlations. So by similar arguments as above, this sum is highly concentrated around its mean 2^{-s}. This gives the number of iterates that we need to make. □

At this point, we have a quantum algorithm that on input z^{in}, creates a "filtered correlation state". The right key in this state has an amplitude proportional to the experimental correlation. It remains to complete the key and check. Recall that to discriminate the right key with certainty, we will perform $r(E)$ trial encryptions.

Lemma 6. *There exists a quantum algorithm* SETUP *such that:*

$$\mathsf{SETUP}\,|0\rangle = 2^{(s-|k^{\mathrm{in}}|)/2} \sum_{z^{\mathrm{in}},z^{\mathrm{out}}\ agreeing} \widehat{\mathrm{cor}}(z^{\mathrm{in}},z^{\mathrm{out}})\,|z^{\mathrm{in}}\rangle\,|z^{\mathrm{out}}\rangle\,|\mathsf{good}\rangle \tag{26}$$

where $\mathsf{good}=1 \iff z^{\mathrm{in}},z^{\mathrm{out}} = k^{\mathrm{in}},k^{\mathrm{out}}$. *Its gate count is:*

$$\mathrm{Tof}\,(\mathsf{SETUP}) = \mathrm{Tof}\,(\mathsf{CORFILT}) + \frac{\pi}{2}2^{|k^c|/2}r(E)\mathrm{Tof}\,(E)\quad . \tag{27}$$

Proof. The algorithm runs as follows. We first create a uniform superposition over z^{in}, then we apply CORFILT, then we use an Exact QAA over the remaining key bits z^c to mark exactly the good subkeys by trial encryptions. We simply run this QAA and set the flag good to 1 if it outputs the good key. Indeed, if $z^{\mathrm{in}},z^{\mathrm{out}} \neq k^{\mathrm{in}},k^{\mathrm{out}}$, the completed key is never good and the QAA returns a uniform superposition of bad keys (so we always write 0). If $z^{\mathrm{in}},z^{\mathrm{out}} = k^{\mathrm{in}},k^{\mathrm{out}}$, we find the good key with certainty, so we always write 1. □

Our algorithm is obtained by applying a QAA on SETUP.

Theorem 1. *Let* $t = 1.005\sqrt{\mathrm{ELP}}2^{n/2}$. *There exists a quantum algorithm that takes no input and returns (after measurement) the master key with probability* $\geq \frac{1}{2}$. *This algorithm has complexity:*

$$\frac{\pi^2}{8t}2^{|K|/2}r(E)\mathrm{Tof}\,(E) + \frac{\pi^2}{8t}2^{(n+|k^{\mathrm{in}}|)/2}\mathrm{Tof}\,(\mathsf{CORCOMP}) \tag{28}$$

and succeeds with probability 0.271 *over the run of the attack.*

Proof. We apply a QAA on SETUP, by amplifying the part which leads to a flag 1, i.e., the good subkey $k^{\mathrm{in}},k^{\mathrm{out}}$. By Equation 23 and the definition of t, with probability 0.271, the corresponding amplitude can be bounded by:

$$t2^{(s-|k^{\mathrm{in}}|-n)/2} \leq 2^{(s-|k^{\mathrm{in}}|)/2}|\widehat{\mathrm{cor}}(k^{\mathrm{in}},k^{\mathrm{out}})| \leq 2t2^{(s-|k^{\mathrm{in}}|-n)/2}\quad . \tag{29}$$

This amplitude determines the number of iterates that we need to perform in the QAA. Given the upper bound, the maximal number that we can apply before over-amplifying is:

$$\left\lfloor \frac{\pi}{4} \left(\arcsin \frac{2t}{2^{(|k^{\text{in}}|-s+n)/2}} \right)^{-1} \right\rfloor \simeq \frac{\pi}{4} \frac{2^{(|k^{\text{in}}|-s+n)/2}}{2t} = \frac{\pi}{8t} 2^{(|k^{\text{in}}|-s+n)/2}$$

The corresponding success probability, i.e., the probability to measure a flag 1 at the end, can be lower bounded as:

$$\sin^2 \left(\left(2\frac{\pi}{8t} 2^{(|k^{\text{in}}|-s+n)/2} + 1 \right) \sqrt{2^{(s-|k^{\text{in}}|)/2} |\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})|} \right) \geq \sin^2 \frac{\pi}{4} = \frac{1}{2} \ .$$

We obtain the complexity as:

$$\frac{\pi}{4t} 2^{(|k^{\text{in}}|-s+n)/2} \text{Tof}\left(\mathsf{SETUP}\right) \ .$$

We replace Tof (SETUP) by its formula and develop to conclude. □

If the first term is dominant, it differs from the complexity of Grover search $(\frac{\pi}{2} 2^{\kappa/2} r \text{Tof}\left(E\right))$ by a factor $\frac{\pi}{4t}$. However Grover search succeeds with overwhelming probability, while this one succeeds only with probability $0.271 \times 0.5 \simeq 0.1355$. The difference in *average* complexity is of a factor: $\frac{\pi}{4t}/0.1355$.

We conclude that we can only use an ELP such that:

$$\frac{\pi}{4}/0.1355 < t = 1.005 \cdot 2^{n/2} \sqrt{\text{ELP}} \implies \text{ELP} \geq 2^{-n+5.06} \ .$$

5 Multiple Linear Cryptanalysis using the QFT

With *multiple* linear approximations, the statistics have a much smaller variance, so it becomes easier to distinguish the right key from the other ones. In this section, we show that this also helps quantum attacks.

5.1 Intuition

We consider the structure of Figure 3, but this time, we introduce M linear approximations: α_i, β_i (we use subscripts to denote the different cases). While the outer key bits k^{out} remain the same in all cases, the inner key bits k_i^{in} can be different, and the number of relations between k_i^{in} and k^{out} (denoted s_i) can vary.

We consider these approximations to be statistically independent. For simplicity, assume for now that all subkey guesses k_i^{in} can be included in a single inner key guess k^{in}. The statistic of interest is denoted $\widehat{q}(k^{\text{in}}, k)$:

$$\widehat{q}(k^{\text{in}}, k) = \sum_{i=1}^{M} \widehat{\text{cor}}_i(k^{\text{in}}, k)^2 \ .$$

For the right subkey, we expect this statistic to be higher. Instead of trying to compute it by hand, we notice already that for individual approximations α_i, β_i, we can obtain the state:

$$|\mathsf{Cor}_{z_i^{\mathrm{in}}}\rangle \simeq \sum_z \widehat{\mathrm{cor}}_i(z^{\mathrm{in}}, z)\,|z\rangle \quad ,$$

where the normalization follows from $\sum_z \widehat{\mathrm{cor}}_i(z^{\mathrm{in}}, z)^2 \simeq 1$. Thus, if we compute this for all M approximations in superposition over i, we obtain:

$$|\mathsf{Cor}_{z^{\mathrm{in}}}\rangle := \frac{1}{\sqrt{M}} \sum_i |\mathsf{Cor}_{z_i^{\mathrm{in}}}\rangle\,|i\rangle = \frac{1}{\sqrt{M}} \sum_{i,z} \widehat{\mathrm{cor}}_i(z^{\mathrm{in}}, z)\,|z\rangle\,|i\rangle \quad . \tag{30}$$

Despite the presence of i, a subsequent QAA layer will only care on the total amplitude that is put on a given key guess. Here, the total amplitude on (z^{in}, z) is $\sqrt{\frac{\widehat{q}(k^{\mathrm{in}},k)}{M}}$, which depends on the multiple linear cryptanalysis statistic. There is no computational overhead, because we *do not compute* the statistic; it simply appears in the amplitude.

5.2 Computation of the Correlation State

We start by computing the state $|\mathsf{Cor}_{z^{\mathrm{in}}}\rangle$ defined in Equation 30 above. This means that:

- The partial computations of the cipher (E', F^R) must now take into account the dependency on i. They might cost more, but remain insignificant;
- The functions f, g in the convolution can depend on i, although we simply write them $f_{z_i^{\mathrm{in}}}, g_{z_i^{\mathrm{in}}}$ to simplify notation;
- We need a quantum circuit GFMULT to compute the Fourier coefficients of these different functions:

$$\mathsf{GFMULT} : |z_i^{\mathrm{in}}\rangle\,|x\rangle\,|i\rangle\,|0\rangle \mapsto |z_i^{\mathrm{in}}\rangle\,|x\rangle\,|i\rangle\,|\widehat{g_{z_i^{\mathrm{in}}}}(x)\rangle \quad .$$

- We need an all-encompassing bound G on the Fourier coefficients of $g_{k_i^{\mathrm{in}}}$, valid for all i simultaneously.

Fortunately in our applications, the different g are actually all similar functions, and there is no big difference in computing their Fourier coefficients.

Lemma 7. *There exists an algorithm* $\mathsf{CORMULT}$ *such that:*

$$\mathsf{CORMULT}\,|z_i^{\mathrm{in}}\rangle\,|i\rangle\,|0\rangle = |z_i^{\mathrm{in}}\rangle\,|i\rangle \sum_z \widehat{\mathrm{cor}}_i(z_i^{\mathrm{in}}, z)\,|z\rangle$$

$$\mathrm{Tof}\,(\mathsf{CORMULT}) = \left(\frac{\pi}{2}\frac{G}{2^{n/2}} + 3\right)(2\mathrm{Tof}\,(\mathsf{GFMULT}) + 2\mathrm{Tof}\,(\mathsf{QUERY})) \quad .$$

Proof. We run Algorithm 2 in superposition over $|i\rangle$. By an analysis similar to Lemma 4, the subroutine outputs:

$$|z_i^{\text{in}}\rangle\,|i\rangle\left(\frac{2^{n/2}}{G}\left(\sum_z \widehat{\text{cor}}_i(z_i^{\text{in}},z)\,|z\rangle\right)|1\rangle+|*\rangle\,|0\rangle\right)\quad . \tag{31}$$

Then, we use an Exact QAA to amplify the component 1. Note that we have simplified the writing by approximating the sum of correlations to 1. □

Before we analyze the amplitude on the right key, we must (like before) use the relations between z_i^{in} and z. These relations differ depending on i, but *there should be the same amount* for all i. This is because the correlations are rescaled by a quantity $2^{s_i/2}$, where s_i is the amount of relations. We can only obtain the statistic $\widehat{q}(z)$ if all the s_i are equal.

Lemma 8. *There exists an algorithm* FILTEREDMULT *such that:*

$$\textsf{FILTEREDMULT}\,|z_i^{\text{in}}\rangle\,|i\rangle\,|0\rangle=|z_i^{\text{in}}\rangle\,|i\rangle\,2^{s/2}\sum_{z_i^{\text{out}}\ agrees}\widehat{\text{cor}}_i(z_i^{\text{in}},z_i^{\text{out}})\,|z_i^{\text{out}}\rangle$$

$$\text{Tof}\,(\textsf{FILTEREDMULT})=\frac{\pi}{2}2^{s/2}\text{Tof}\,(\textsf{CORMULT})\quad .$$

Proof. On the output of CORMULT, we apply an Exact QAA. □

5.3 QAA Layers

With the same structure as in Section 4.3, we add other QAA layers to complete our algorithm.

We start by completing the subkeys. Like before, we use an Exact QAA which marks the key guesses $(z_i^{\text{in}},z_i^{\text{out}})=(k_i^{\text{in}},k_i^{\text{out}})$. The resulting algorithm is the "setup" on which we will apply QAA again. Notice that from now on, since all keys have the same amount of relations, we have $\forall i,j,|k_i^{\text{in}}|=|k_j^{\text{in}}|,|k_i^c|=|k_j^c|$.

Lemma 9. *There exists an algorithm* SETUPMULT *such that:*

$$\textsf{SETUPMULT}\,|0\rangle=\sum_{i,z_i^{\text{in}},z_i^{\text{out}}\ agree}\frac{2^{(s-|k^{\text{in}}|)/2}}{\sqrt{M}}\widehat{\text{cor}}_i(z_i^{\text{in}},z_i^{\text{out}})\,|i\rangle\,|z_i^{\text{in}},z_i^{\text{out}}\rangle\,|\textsf{good}\rangle$$

$$\text{Tof}\,(\textsf{SETUPMULT})\leq\text{Tof}\,(\textsf{FILTEREDMULT})+\frac{\pi}{2}2^{|k^c|/2}r(E)\text{Tof}\,(E)$$

Proof. The algorithm runs as follows. We first create a uniform superposition over i and z_i^{in}, then we apply FILTEREDMULT, then we mark the good subkeys with an Exact QAA. There are $\frac{\pi}{4}2^{|k^c|/2}$ iterates; each iterate calls E a total of $2r(E)$ times. □

Finally, we apply QAA on top of this algorithm, where we just want to find a subkey guess $z_i^{\text{in}},z_i^{\text{out}}$ (whichever the i) which completes into the good key.

Theorem 2. *Assuming $M \gg 1$, there exists an algorithm which outputs a good guess $i, k_i^{\text{in}}, k_i^{\text{out}}$, with probability of success $\geq 1 - \frac{18}{M}$, in time:*

$$T := \frac{1}{\sqrt{2^n C/M + 1}} \left(\frac{\pi^2}{4} 2^{|K|/2} r \text{Tof}\,(E) + \frac{\pi^2}{4} 2^{(n+|k^{\text{in}}|)/2} \text{Tof}\,(\mathsf{CORMULT}) \right) . \tag{32}$$

Proof. On the output of SETUPMULT, the total probability of measuring the flag 1, which corresponds exactly to the good subkeys for different paths, is equal to:

$$p := \sum_i \frac{2^{s-|k^{\text{in}}|}}{M} \widehat{\text{cor}}_i(k_i^{\text{in}}, k_i^{\text{out}})^2 = \frac{2^{s-|k^{\text{in}}|}}{M} \widehat{q}(k^{\text{in}}, k^{\text{out}}) \ . \tag{33}$$

By the right-key randomization hypothesis, $\widehat{q}(k^{\text{in}}, k^{\text{out}})$ follows a normal distribution with mean $C + M2^{-n}$ and variance $\frac{2}{M}C^2$ where C is the capacity of the multiple approximation. Thus, with overwhelming probability, we have:

$$C\left(1 - \frac{3\sqrt{2}}{\sqrt{M}}\right) + M2^{-n} \leq \widehat{q}(k^{\text{in}}, k^{\text{out}}) \leq C\left(1 + \frac{3\sqrt{2}}{\sqrt{M}}\right) + M2^{-n} \tag{34}$$

Thus, we can bound $p'(1-\varepsilon) \leq p \leq p'(1+\varepsilon)$ where $p' = 2^{s-|k^{\text{in}}|}\left(\frac{C}{M} + 2^{-n}\right)$ and $\varepsilon = \frac{3\sqrt{2}}{\sqrt{M}}$. We conclude with Lemma 2. □

If the first term in Equation 32 is dominant, then we can have an advantage with respect to Grover search. The main interest in using the multiple cryptanalysis statistic is that we have reduced the variance on the right key case, allowing a good probability of success for this procedure (instead of the 0.1355 given by Theorem 1).

6 Applications

In this section, we give several examples of our technique. We start with the block ciphers FLY [35] and PIPO [36]. In both cases our QFT-based algorithm can reach more rounds than previous quantum linear attacks, like classical FFT-based cryptanalysis. We assume either Q2 queries (in which case they are not the dominant cost anyway), or QRACM queries, with a QRACM containing the whole codebook. Because we use large ELPs in both cases, it would be possible to reduce the data complexity (but this complicates the analysis).

6.1 Linear Characteristics on FLY and PIPO

Both ciphers have similar structures: a 64-bit state, S-Boxes of 8 bits, a linear layer which is a simple bit permutation (like PRESENT), and a trivial key schedule. FLY is defined only with 128-bit keys, with 20 rounds. The round function is represented on Figure 4: it applies a round key addition, followed by 8 parallel

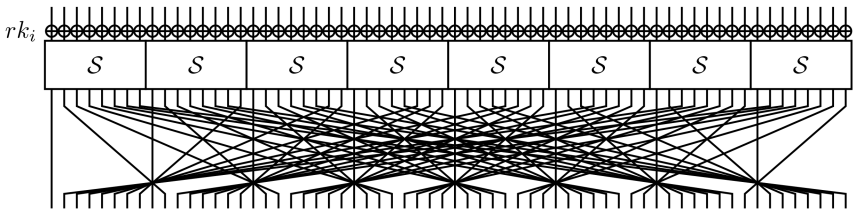

Fig. 4. FLY round function (Figure 5.2 in [35].

S-Boxes of 8 bits (the S-Box LITTLUN-1 which is also defined in [35]) and a permutation of the bits.

We consider the simple key-schedule KS1 proposed for a case where related-key security is not required. The master key K of 128 bits is divided into two halves: $K = k_0 \| k_1$ and the round keys alternate between k_0 and $k_0 \oplus k_1$. In the following, we replace $(k_0, k_0 \oplus k_1)$ by (k_0, k_1), since this makes no difference. For other details of the specification which are irrelevant for our attack (such as round constants) we refer to [35].

The bit-permutation in PIPO is different, but it has the same property that the 8 bits of each S-Box are distributed to the 8 S-Boxes of the next round. There are two versions: PIPO-128 and PIPO-256 with respectively 13 and 17 rounds (i.e., 14 and 18 subkey additions). In both cases, the master key K is cut into $|K|/64$ subkeys which are XORed alternatively to the state. The authors [36] claimed attacks for up to 9 / 13 and 11 / 17 rounds, but without details. In [31], quantum circuits for PIPO were given, using respectively $1248 = 2^{10.29}$ and $1632 = 2^{10.67}$ Toffoli gates for the two versions. For FLY, a quick look at its 8-bit S-Box shows that it contains 12 nonlinear operations (AND and OR) which can be implemented with 12 Toffoli gates. For 11 complete rounds this gives the count of $8 \times 11 \times 24 = 2112 = 2^{11.04}$ Toffoli gates.

Search for Linear Distinguishers. In both cases, we follow the approach of [1] to search for linear trails in a restricted family of masks. We use all masks which activate at most 2 S-Boxes, and at most 2 S-Boxes before the previous permutation layer. This forms a family of 6000 masks. We compute the 6000×6000 sparse correlation matrix through a single round, then obtain the ELPs for multiple rounds via matrix multiplication. In our attacks, we only use trails that activate *exactly* two S-Boxes at the rounds before and after.

Results (Table 1). On FLY, we find 64 approximations through 8 full rounds having ELP of order 2^{-58}. We did not find any useful approximation through 9 rounds (ELPs are too close to 2^{-64}). On PIPO, we find 24 approximations through 5 rounds of ELP 2^{-50}. Although this is the best ELP for 5 rounds, we prefer having more suboptimal approximations, so we rely on a family of 112 approximations of ELPs between $2^{-51.8}$ and 2^{-52}. Likewise, we did not find any

Table 1. Results on linear characteristics.

Cipher	Rounds	M	ELP
Fly	8	64	2^{-58}
Pipo	5	112	$\geq 2^{-52}$
Pipo	5	24	2^{-50}

useful approximations for 6 rounds. Nevertheless, the authors of [36] report a characteristic on 6 rounds using the branch-and-bound technique, so it is likely that one can improve over our estimates in both cases.

6.2 Attacks on FLY and PIPO-128

On Fly and Pipo-128, which have the same structure, we can propose a similar attack pattern. Following Figure 3, we remove the block E' and append two rounds before (this corresponds to F_L) and one round after (F_R) the distinguisher.

In the case of Fly, the distinguisher contains 8 rounds, so the outer key k^{out} is k_0 and the inner key k^{in} contains $\leq 16 + 16$ bits of k_1 (because only two S-Boxes are active in these rounds). Whichever the linear approximation considered, due to the regularity of the bit permutation, there are exactly bit relations between the 16 bits of k_1 in the second and last rounds, so $|k^{\text{in}}| = 28$.

In the case of Pipo, we have the same scheduling but the distinguisher spans 5 rounds, so the outer key is k_0 and the inner key contains exactly 16 bits of k_0 and 16 bits of k_1. We have $|k^{\text{in}}| = 32$ and $s = 16$ bits of relation between k^{in} and k^{out}.

In both cases, we use multiple linear cryptanalysis and Theorem 2. Using M linear approximations with capacity C, our algorithm succeeds with probability $1 - \frac{18}{M}$ and runs in time:

$$T = \frac{1}{\sqrt{2^n C/M + 1}} \left[\frac{\pi^2}{4} 2^{|K|/2} r(E) T(E) + \frac{\pi^2}{4} 2^{\frac{n+|k^{\text{in}}|}{2}} \text{Tof}\,(\mathsf{CORMULT}) \right] . \quad (35)$$

We take $r(E) = 3$, as 3 plaintext-ciphertext pairs are enough for checking the key.

Computing the Fourier Coefficients. We must implement the circuit GFMULT, which computes $\widehat{g_{z_i^{\text{in}}}}(x) = \sum_y (-1)^{x \cdot y} (-1)^{\beta \cdot F^L_{z^{\text{in}}}(y)}$ in superposition over z^{in} (current inner key guess), i and x. We show how to reduce this to a feasible computation, using the structure of F^L. We expect that this will also be possible in general for a small number of rounds in any SPN cipher.

All linear approximations considered activate two S-Boxes before the distinguisher. Thus, if we cut the input x into $x_0, \ldots, x_7$ and the inner key z into bits

$z_0, z_1, \ldots, z_{15}$, we can simplify $F_z^L(x)$ into:

$$F_z^L(x) = S'_0\left[z_0 + S_0(x_0), z_1 + S_1(x_1), \ldots, z_7 + S_7(x_7)\right], \\ S'_1\left[z_8 + S_8(x_0), z_9 + S_9(x_1), \ldots, z_{15} + S_{15}(x_7)\right] \quad (36)$$

where all these functions are simply applying S-Boxes and selecting some input and output bits, up to permutations of these bits.

By merging S'_0 and S'_1 with the scalar product on β, we can rewrite this as:

$$g_z(x) = S'_0\left[z_0 + S_0(x_0), z_1 + S_1(x_1), \ldots, z_7 + S_7(x_7)\right] \times \\ S'_1\left[z_8 + S_8(x_0), z_9 + S_9(x_1), \ldots, z_{15} + S_{15}(x_7)\right] \quad (37)$$

where S'_0 and S'_1 are functions into $\{-1, 1\}$. The independence between these different parts is the key to a faster computation of $\widehat{g_z}$. Indeed:

$$\begin{aligned}
\widehat{g_z}(y) &= \sum_x (-1)^{x\cdot y} g_z(x) \\
&= \sum_{u_0,\ldots,u_{15}} \sum_{\substack{x_0 | z_0 + S_0(x_0) = u_0 \\ z_8 + S_8(x_0) = u_8}} \cdots \sum_{\substack{x_7 | z_7 + S_7(x_7) = u_7 \\ z_{15} + S_{15}(x_7) = u_{15}}} (-1)^{x\cdot y} g_z(x) \\
&= \sum_{u_0,\ldots,u_{15}} \prod_{i=0}^{7} \underbrace{\left(\sum_{\substack{x_i \\ S_i(x_i) = u_i \\ S_{i+8}(x_i) = u_{i+8}}} (-1)^{x_i \cdot y_i} \right)}_{:= h_i(y_i, u_i, u_{i+8})} \begin{array}{l} S'_0(u_0 + z_0, \ldots, u_7 + z_7) \\ \times S'_1(u_8 + z_8, \ldots, u_{15} + z_{15}) \end{array} .
\end{aligned}$$

At this point, we already arrive at a feasible cost using some precomputations (and QRACM tables). But we can reduce this cost further by noticing that it is easier to compute $\widehat{g_z}$ *directly into the amplitude* than digitally.

Amplitude Computation Pattern. We start from the state:

$$\sum_y \widehat{f_z}(y) \left|y\right\rangle$$

We first append a 16-bit value u, in uniform superposition:

$$\sum_y \sum_u \widehat{f_z}(y) \left|y\right\rangle \left|u\right\rangle \quad .$$

We compute digitally the functions h_i, their product, and both S'_0 and S'_1:

$$\sum_y \sum_u \widehat{f_z}(y) \left|y\right\rangle \left|u\right\rangle \left|S'_0(u+z) S'_1(u+z) \prod_i h_i(y_i, u_i, u_{i+8})\right\rangle \quad .$$

If we consider that the products of h_i are a sum of $(2^6)^8 = 2^{48}$ independent variables taking value ± 1, then by the argument of Lemma 10, the value to multiply in the amplitude is upper bounded by $2^{24}H$ where:

$$H := \sqrt{6(\ln 100 + 49 \ln 2 + \ln M)} \simeq \sqrt{2^{7.85} + 6 \ln M} \ .$$

After doing the product in the amplitude, we obtain a quantum state of the form:

$$\alpha \sum_y \sum_u \widehat{f}_z(y) \left(\prod_{i=0}^{7} h_i(y_i, u_i, u_{i+8}) S'_0(u+z) S'_1(u+z) \right) |y\rangle |u\rangle |0\rangle + |\Phi\rangle$$

where $|\Phi\rangle$ is a non-normalized bad state and α is such that we project on 0 with probability about $\frac{1}{H^2}$.

Next, we apply a Hadamard transform on the u register. The "0" component of the state evolves as follows:

$$\frac{\alpha}{\sqrt{2^{16}}} \sum_y \sum_{u,v} \widehat{f}_z(y) \left(\prod_{i=0}^{7} h_i(y_i, u_i, u_{i+8}) S'_0(u+z) S'_1(u+z) \right) (-1)^{u \cdot v} |y\rangle |v\rangle \ .$$

Thus, the component $v = 0$ has amplitude about $\frac{1}{H2^8}$, and it corresponds to the state $\sum_y \widehat{f}_z(y)\widehat{g_z}(y) |y\rangle$ that we want. In total, we can create the correlation state via a procedure which performs $\frac{\pi}{4} H 2^8$ iterates of QAA. Each iterate queries f_z twice and does the following operations: compute the h_i (8×2^8 S-Box computations), their product (7 multiplications of 32-bit integers, i.e., 1584×7 Toffolis) and S_0, S'_1 (2 S-Box computations) twice. Using a Toffoli count of 12 for the Fly S-Box (less for the Pipo S-Box), this gives an additional gate count of $35688 = 2^{15.12}$ Toffolis, and:

$$\text{Tof}\,(\mathsf{CORMULT}) = \frac{\pi}{2} 2^8 \sqrt{2^{7.85} + 6 \ln M} \left(\text{Tof}\,(\mathsf{QUERY}) + 2^{15.12} \right) \ . \tag{38}$$

In both our attacks we have $M \leq 112$ so we can write: $\text{Tof}\,(\mathsf{CORMULT}) \leq 2^{27.78}$. We will also consider that $\text{Tof}\,(\mathsf{QUERY})$ is dominated by the second term. Indeed, it performs either a query to the cipher (around 2^{11} Toffolis) or a QRACM query.

Results. For Pipo, we have $M = 112$ and $C = M \times 2^{-52}$, thus:

$$\begin{aligned} \text{Tof} &= 2^{-6.00} \times \left(2^{2.89} \times 2^{64} \times 2^{10.29} + 2^{1.30} \times 2^{32+16} \times 2^{27.78} \right) \\ &= 2^{-6.00} \left(2^{77.18} + 2^{77.08} \right) = 2^{72.13} \ . \end{aligned}$$

The success probability is $1 - \frac{18}{M} = 0.84$. This compares favorably to an exhaustive search of the key in T-gate count $2^{74.94}$ (multiplying the count of [36] for Pipo-128 with a factor $\frac{\pi}{2} 2^{64}$).

For Fly, we have a smaller capacity $C = M \times 2^{-58}$, but benefit from 4 bit-relations which reduce a little the complexity. Furthermore, there is one less

level of QAA since there are no relations to enforce between k^{in} and k^{out}.

$$\begin{aligned}\text{Tof} &= 2^{-3.01} \times \left(2^{2.89} \times 2^{64} \times 2^{10.67} + 2^{0.65} \times 2^{32+14} \times 2^{27.78}\right) \\ &= 2^{-3.01} \left(2^{77.56} + 2^{74.43}\right) = 2^{74.71} \ .\end{aligned}$$

These results are summarized in Table 2. In both cases, the quantum linear attacks from [34] are applicable, but reach one less round.

Table 2. Summary of attacks. "Data" in the Q2 setting is the total number of Q2 queries performed during the attack.

Attack	Rounds	Time (Toffoli gates)	Success prob.	Data	Memory
		PIPO-128			
Classical search	13 / 13	2^{128}	1	3	negl.
Quantum search (Q1)	13 / 13	$2^{74.94}$	1	3	negl.
Linear QFT (Q1)	8 / 13	$2^{72.13}$	0.84	2^{64}	2^{71} QRACM
Linear QFT (Q2)	8 / 13	$2^{72.13}$	0.84	$2^{61.96}$	negl.
		FLY			
Classical search	20 / 20	2^{128}	1	3	negl.
Quantum search (Q1)	20 / 20	$2^{75.69}$	1	3	negl.
Linear QFT (Q1)	11 / 20	$2^{74.11}$	0.72	2^{64}	2^{71} QRACM
Linear QFT (Q2)	11 / 20	$2^{74.11}$	0.72	$2^{59.31}$	negl.

6.3 Discussion on Other Applications

We have tried to apply our technique to other block ciphers, but ran into the limitations of our framework.

Present. For PRESENT, linear attacks using the FWHT give the best results classically (up to 29 rounds out of 31 for the 128-bit key version [22]). However, we quickly run into the following problem: the simple quantum linear attack (without QFT) can work with relatively smaller ELPs, while our attack needs relatively bigger ones (e.g., $2^{-n+5.06}$ for Theorem 1).

Thus, if we try to use our framework to add a round of key-recovery, we lose roughly one round in the linear distinguisher: we cannot demonstrate the interest of this technique on PRESENT with a single linear approximation.

With multiple approximations, the increase in amplitude in the correlation state is also so small that it becomes difficult to observe any speedup.

NOEKEON. We studied the linear attack on NOEKEON given in [15]. Here the key and blocks have the same length of 128 bits. The classical data complexity

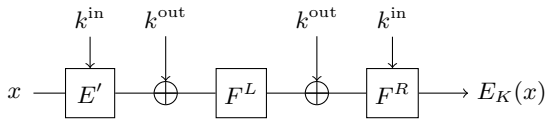

Fig. 5. Generic Structural Attack.

being at least around 2^{120}, the only way to compete with Grover search is the Q2 model. Here the formula of Theorem 1 becomes:

$$\frac{\pi^2}{8t}2^{128/2}r(E)\mathrm{Tof}\,(E)+\frac{\pi^2}{8t}2^{128/2}\mathrm{Tof}\,(\mathsf{CORCOMP})\quad . \tag{39}$$

However, the computation of the correlation state is always more costly than a single query to the cipher, and the factor t does not compensate this enough. Like in the case of PRESENT, the ELP would need to be larger than what the standard linear attack can use.

6.4 Structural Attacks and a Beyond-quadratic Speedup

Replacing E_M by a key addition in Figure 3, and removing the inner key k^{in} from F^L, we obtain the structure represented in Figure 5. Here, our algorithm can achieve a better speedup than quadratic, up to 2.5 precisely, like the offline-Simon algorithm [13]. On a construction like this, offline-Simon needs at least $2^n(1-\mathcal{O}(1/n))$ classical known-plaintext queries (Lemma 1 in [13]). However, our algorithm can use *any* number of known-plaintext queries. Its cost is dominated by QRACM queries.

Theorem 3. *Let t be the gate cost of a QRACM query and a cipher evaluation. Given N classical known-plaintext queries to the E_K of Figure 5, there exists a quantum algorithm recovering K in $\mathcal{O}\left(2^{|k^{\text{in}}|/2}\frac{2^{n/2}}{\sqrt{N}}\sqrt{n}(n+t)+n2^n\right)$ gates, using $\mathcal{O}(n2^n)$ bits of QRACM.*

Proof. We follow the analysis in Section 4.2, using an arbitrary boolean mask $\alpha=\beta=(1,0,\ldots,0)$. Because there is no dependency on z^{in} in the middle permutation Π, we can precompute its Walsh-Hadamard transform in $\mathcal{O}(n2^n)$ and store it in $\mathcal{O}(n2^n)$ bits of QRACM. For the good key k^{in}, the wrong experimental correlations are of order $\mathcal{O}\left(\frac{\sqrt{N}}{2^n}\right)$, while the right key guess reaches exactly $\frac{N}{2^n}$: indeed, the correlation of a linear function is 1. Computing the correlation state takes $\mathcal{O}(\sqrt{n})$ iterates (because of the bound on Fourier coefficients for a random function of Lemma 10) of a procedure using $\mathcal{O}(n+t)$ gates (a comparator and a query to the QRACM).

After creating the correlation state for a given z^{in}, we complete the setup by performing a trial encryption. With a uniform superposition over z^{in}, the right key $k^{\text{in}},k^{\text{out}}$, marked with 1, has an amplitude equal to: $\frac{1}{2^{|k^{\text{in}}|/2}}\times\frac{1}{\sqrt{N\times 2^n}}N=$

$\mathcal{O}\left(\sqrt{N}2^{-(|k^{\text{in}}|+n)/2}\right)$. Therefore, the key is found using $\mathcal{O}\left(2^{(|k^{\text{in}}|+n)/2}/\sqrt{N}\right)$ iterates of QAA. □

In particular, if QRACM queries are considered as costly as block cipher evaluations (typically $\mathcal{O}(n^2)$ gates), then the gate count at the minimal point $N = 2^n(1 - \mathcal{O}(1/n))$ is *smaller* than the one of offline-Simon, which requires n block cipher calls and $\mathcal{O}(n^3)$ gates per iterate.

7 Conclusion and Open Problems

In this paper, we have introduced a new technique in quantum key-recovery attacks on block ciphers. After Hosoyamada [28] showed that one could use the Quantum Fourier Transform in a statistical cryptanalysis, this technique shows that we can use it in key-recovery attacks. From the perspective of quantum algorithms, we have switched from Simon's algorithm, which is limited to strong algebraic structures, to computing a discrete convolution. However, this new perspective opens several important questions.

Computing Fourier Coefficients. While the construction of the *correlation state* is central to our work, it is also quite technical, due to the computation of Fourier coefficients into the amplitudes of a quantum state. In our applications, we have shown that this could be done efficiently by considering the structure of the functions involved. As a future work, we plan to give a generic algorithm and complexity analysis for the relevant cases in FFT-based linear cryptanalysis, e.g., a small number of rounds of any SPN structure. However, more generally, we do not know if there exists a competitive generic algorithm for this task.

Problem 1. Let $f, g : \mathbb{F}_2^n \to \{-1, 1\}$ be two functions. Given query access to g, given a black box that produces $\sum_x \widehat{f}(x)\,|x\rangle$, produce $\sum_x \widehat{g}(x)\widehat{f}(x)\,|x\rangle$.

Assuming that the Fourier coefficients of g and f are distributed somewhat uniformly, we can produce the state $\sum_x \widehat{g}(x)\widehat{f}(x)\,|x\rangle$ with $\mathcal{O}\left(2^{n/2}\right)$ queries to both functions: we simply start with $\sum_x \widehat{g}(x)\,|x\rangle \sum_y \widehat{f}\,|y\rangle$ and amplify the part of the state where $x = y$. But if we use this in Theorem 1, we will need at least $|K| \geq 2n$ to obtain a speedup with respect to exhaustive search, leaving this generic method useless for most applications.

Finding the Largest Correlation. After building the correlation state, we need to find the key which has the largest experimental correlation. In general, the problem that we would like to solve is the following.

Problem 2. Given a black-box quantum that produces a state $\sum_x \alpha_x\,|x\rangle$, where the amplitude are distributed according to a centered Gaussian, and either: • all of them are below a threshold t; • exactly one of them is above the threshold t; determine the case and / or find the corresponding coordinate.

This problem is not specific to symmetric cryptanalysis, as it appears in quantum algorithms for dual lattice sieving attacks [3]. In the worst case, the experimental correlations have a standard deviation of $\mathcal{O}\left(2^{-n/2}\right)$ and the largest one is only of order $\mathcal{O}\left(2^{n/2}\right)$ as well. Here, we do not know of any algorithm faster than $\mathcal{O}(2^n)$. Unfortunately, this case seems typical in both applications (lattice sieving and linear cryptanalysis).

A related problem is *zero-correlation* attacks, where the experimental correlation of the right subkey, instead of being bigger, is exactly zero.

Problem 3. Given a black-box quantum that produces a state $\sum_x \alpha_x |x\rangle$, where the amplitude are distributed according to a centered Gaussian, and either: • one of them is exactly zero; • or not; determine the case and / or find the corresponding coordinate.

Again, when the others have a standard deviation $\mathcal{O}\left(2^{-n/2}\right)$, no algorithm better than $\mathcal{O}(2^n)$ is known. Consequently, we do not know how to exploit this property, which would be very useful for key-recovery attacks.

Acknowledgments. The author thanks Xavier Bonnetain, Antonio Flórez-Gutiérrez and María Naya-Plasencia for helpful discussions and comments.

Appendix

A Bounding Fourier Coefficients

Lemma 10. *Let $f_i : \{0,1\}^n \to \{-1,1\}$, $1 \leq i \leq M$ be a family of independent random functions. With probability at least 0.99, it holds that:*

$$\forall z, \forall i, |\widehat{f_i}(z)| \leq 2^{n/2}\sqrt{6(\ln 100 + (n+1)\ln 2 + \ln M)} \ . \tag{40}$$

Proof. Let $f : \{0,1\}^n \to \{-1,1\}$ be a random function. We want to bound the maximum of its Fourier coefficients: $\max_z |\widehat{f}(z)|$.

We consider each coefficient separately, although they are not independent. For each z, $\widehat{f}(z)$ is a random variable over f equal to: $2\mathrm{Bin}(2^n, 1/2) - 2^n = 2\left(\mathrm{Bin}(2^n, 1/2) - 2^{n-1}\right)$. We use a Chernoff bound:

$$\begin{aligned}
\forall \delta, \forall z, \Pr_f(|\mathrm{Bin}(2^n, 1/2) - 2^{n-1}| \geq \delta 2^{n-1}) &\leq 2\exp\left(\frac{-\delta^2 2^n}{6}\right) \\
\Pr_f(|\widehat{f}(z)| \geq \delta 2^n) &\leq 2\exp\left(\frac{-\delta^2 2^n}{6}\right) \\
\implies \forall \delta, \forall z, \Pr_f(|\hat{g}(z)| \geq \delta\sqrt{2^n}) &\leq 2\exp\left(\frac{-\delta^2}{6}\right) \\
\implies \forall \delta, \Pr_f(\exists z, |\hat{g}(z)| \geq \delta\sqrt{2^n}) &\leq 2^{n+1}\exp\left(\frac{-\delta^2}{6}\right) \ .
\end{aligned}$$

We find a value of δ for which this probability is smaller than $1/100$:

$$\ln(2^{n+1}) - \frac{\delta^2}{6} \leq -\ln 100 \implies \delta \geq \sqrt{6(\ln 100 + (n+1)\ln 2)} \ .$$

References

1. Abdelraheem, M.A.: Estimating the probabilities of low-weight differential and linear approximations on present-like ciphers. In: ICISC. Lecture Notes in Computer Science, vol. 7839, pp. 368–382. Springer (2012)
2. Alagic, G., Bai, C., Katz, J., Majenz, C.: Post-quantum security of the even-mansour cipher. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 13277, pp. 458–487. Springer (2022)
3. Albrecht, M.R., Shen, Y.: Quantum augmented dual attack. IACR Cryptol. ePrint Arch. p. 656 (2022)
4. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.V.: Strengths and weaknesses of quantum computing. SIAM J. Comput. **26**(5), 1510–1523 (1997)
5. Bernstein, E., Vazirani, U.V.: Quantum complexity theory. SIAM J. Comput. **26**(5), 1411–1473 (1997)
6. Biryukov, A., Cannière, C.D., Quisquater, M.: On multiple linear approximations. In: CRYPTO. Lecture Notes in Computer Science, vol. 3152, pp. 1–22. Springer (2004)
7. Blondeau, C., Nyberg, K.: Improved parameter estimates for correlation and capacity deviates in linear cryptanalysis. IACR Trans. Symmetric Cryptol. **2016**(2), 162–191 (2016)
8. Blondeau, C., Nyberg, K.: Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. Des. Codes Cryptogr. **82**(1-2), 319–349 (2017)
9. Bogdanov, A., Geng, H., Wang, M., Wen, L., Collard, B.: Zero-correlation linear cryptanalysis with FFT and improved attacks on ISO standards camellia and CLEFIA. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 8282, pp. 306–323. Springer (2013)
10. Bogdanov, A., Rijmen, V.: Linear hulls with correlation zero and linear cryptanalysis of block ciphers. Des. Codes Cryptogr. **70**(3), 369–383 (2014)
11. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: The offline simon's algorithm. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 11921, pp. 552–583. Springer (2019)
12. Bonnetain, X., Naya-Plasencia, M., Schrottenloher, A.: Quantum security analysis of AES. IACR Trans. Symmetric Cryptol. **2019**(2), 55–93 (2019)
13. Bonnetain, X., Schrottenloher, A., Sibleyras, F.: Beyond quadratic speedups in quantum attacks on symmetric schemes. In: EUROCRYPT (3). Lecture Notes in Computer Science, vol. 13277, pp. 315–344. Springer (2022)
14. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002)
15. Broll, M., Canale, F., Flórez-Gutiérrez, A., Leander, G., Naya-Plasencia, M.: Generic framework for key-guessing improvements. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 13090, pp. 453–483. Springer (2021)
16. Collard, B., Standaert, F., Quisquater, J.: Improving the time complexity of matsui's linear cryptanalysis. In: ICISC. Lecture Notes in Computer Science, vol. 4817, pp. 77–88. Springer (2007)
17. Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. J. Math. Cryptol. **1**(3), 221–242 (2007)
18. van Dam, W., Hallgren, S., Ip, L.: Quantum algorithms for some hidden shift problems. SIAM J. Comput. **36**(3), 763–778 (2006)

19. Davenport, J.H., Pring, B.: Improvements to quantum search techniques for block-ciphers, with applications to AES. In: SAC. Lecture Notes in Computer Science, vol. 12804, pp. 360–384. Springer (2020)
20. David, N., Naya-Plasencia, M., Schrottenloher, A.: Quantum impossible differential attacks: Applications to AES and SKINNY. IACR Cryptol. ePrint Arch. p. 754 (2022)
21. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **439**(1907), 553–558 (1992)
22. Florez Gutierrez, A.: Optimising Linear Key Recovery Attacks with Affine Walsh Transform Pruning. In: ASIACRYPT 2022 - 28th Annual International Conference on the Theory and Application of Cryptology and Information Security. Taipei, Taiwan (Dec 2022), https://hal.inria.fr/hal-03878737
23. Flórez-Gutiérrez, A., Naya-Plasencia, M.: Improving key-recovery in linear attacks: Application to 28-round PRESENT. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 12105, pp. 221–249. Springer (2020)
24. Frixons, P., Naya-Plasencia, M., Schrottenloher, A.: Quantum boomerang attacks and some applications. In: SAC. Lecture Notes in Computer Science, vol. 13203, pp. 332–352. Springer (2021)
25. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: STOC. pp. 212–219. ACM (1996)
26. Grover, L.K.: Synthesis of quantum superpositions by quantum computation. Physical review letters **85**(6), 1334 (2000)
27. Häner, T., Roetteler, M., Svore, K.M.: Optimizing quantum circuits for arithmetic. arXiv preprint arXiv:1805.12445 (2018)
28. Hosoyamada, A.: Quantum speed-up for multidimensional (zero correlation) linear and integral distinguishers. Cryptology ePrint Archive, Paper 2022/1558 (2022), https://eprint.iacr.org/2022/1558, https://eprint.iacr.org/2022/1558
29. Hosoyamada, A., Sasaki, Y.: Quantum demiric-selçuk meet-in-the-middle attacks: Applications to 6-round generic feistel constructions. In: SCN. Lecture Notes in Computer Science, vol. 11035, pp. 386–403. Springer (2018)
30. Hosoyamada, A., Sasaki, Y.: Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 12106, pp. 249–279. Springer (2020)
31. Jang, K., Song, G., Kwon, H., Uhm, S., Kim, H., Lee, W.K., Seo, H.: Grover on pipo. Electronics **10**(10), 1194 (2021)
32. Jr., B.S.K., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: CRYPTO. Lecture Notes in Computer Science, vol. 839, pp. 26–39. Springer (1994)
33. 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)
34. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symmetric Cryptol. **2016**(1), 71–94 (2016)
35. Karpman, P., Grégoire, B.: The littlun s-box and the fly block cipher. In: Lightweight Cryptography Workshop (2016)
36. Kim, H., Jeon, Y., Kim, G., Kim, J., Sim, B., Han, D., Seo, H., Kim, S., Hong, S., Sung, J., Hong, D.: PIPO: A lightweight block cipher with efficient higher-order masking software implementations. In: ICISC. Lecture Notes in Computer Science, vol. 12593, pp. 99–122. Springer (2020)

37. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In: TQC. LIPIcs, vol. 22, pp. 20–34. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2013)
38. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: ISIT. pp. 2682–2685. IEEE (2010)
39. Kuwakado, H., Morii, M.: Security on the quantum-type even-mansour cipher. In: ISITA. pp. 312–316. IEEE (2012)
40. Leurent, G., Pernot, C., Schrottenloher, A.: Clustering effect in simon and simeck. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 13090, pp. 272–302. Springer (2021)
41. Matsui, M.: Linear cryptanalysis method for DES cipher. In: EUROCRYPT. Lecture Notes in Computer Science, vol. 765, pp. 386–397. Springer (1993)
42. Matsui, M.: The first experimental cryptanalysis of the data encryption standard. In: CRYPTO. Lecture Notes in Computer Science, vol. 839, pp. 1–11. Springer (1994)
43. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
44. NIST: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf
45. Sanders, Y.R., Low, G.H., Scherer, A., Berry, D.W.: Black-box quantum state preparation without arithmetic. Physical review letters **122**(2), 020502 (2019)
46. Selinger, P.: Efficient clifford+ t approximation of single-qubit operators. arXiv preprint arXiv:1212.6253 (2012)
47. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: FOCS. pp. 124–134. IEEE Computer Society (1994)
48. Simon, D.R.: On the power of quantum computation. SIAM J. Comput. **26**(5), 1474–1483 (1997)
49. Sun, L., Wang, W., Wang, M.: Improved attacks on GIFT-64. In: SAC. Lecture Notes in Computer Science, vol. 13203, pp. 246–265. Springer (2021)
50. Takahashi, Y., Tani, S., Kunihiro, N.: Quantum addition circuits and unbounded fan-out. arXiv preprint arXiv:0910.2530 (2009)
51. Yamakawa, T., Zhandry, M.: Verifiable quantum advantage without structure. In: FOCS. pp. 69–74. IEEE (2022)