Improved Quantum Linear Attacks and Application to CAST

Kaveh Bashiri[1], Xavier Bonnetain[2], Akinori Hosoyamada[3], Nathalie Lang[4] and André Schrottenloher[5]

[1] Bundesamt für Sicherheit in der Informationstechnik (BSI), Germany, firstname.lastname@bsi.bund.de
[2] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France, firstname.lastname@inria.fr
[3] NTT Social Informatics Laboratories, Tokyo, Japan
NTT Research Center for Theoretical Quantum Information, Atsugi, Japan
akinori.hosoyamada@ntt.com
[4] Bauhaus-Universität Weimar, Weimar, Germany, nathalie.lang@uni-weimar.de
[5] Univ Rennes, Inria, CNRS, IRISA, Rennes, France, firstname.lastname@inria.fr

Abstract. This paper studies quantum linear key-recovery attacks on block ciphers. The first such attacks were last-rounds attacks proposed by Kaplan et al. (ToSC 2016), which combine a linear distinguisher with a guess of a partial key. However, the most efficient classical attacks use the framework proposed by Collard et al. (ICISC 2007), which computes experimental correlations using the Fast Walsh-Hadamard Transform. Recently, Schrottenloher (CRYPTO 2023) proposed a quantum version of this technique, in which one uses the available data to create a quantum *correlation state*, which is a superposition of subkey candidates where the amplitudes are the corresponding correlations. A limitation is that the good subkey is not marked in this state, and cannot be found easily.
In this paper, we combine the correlation state with another distinguisher. From here, we can use Amplitude Amplification to recover the right key. We apply this idea to Feistel ciphers and exemplify different attack strategies on LOKI91 before applying our idea on the CAST-128 and CAST-256 ciphers. We demonstrate the approach with two kinds of distinguishers, quantum distinguishers based on Simon's algorithm and linear distinguishers. The resulting attacks outperform the previous Grover-meet-Simon attacks.

Keywords: Quantum cryptanalysis · Linear cryptanalysis · Fast Fourier Transform · CAST

1 Introduction

In the past decade, many studies have shown that classical cryptanalysis techniques can be adapted to the quantum setting, such as differential and linear attacks [KLLN16b], rebound attacks [HS20], (Demirci-Selçuk) meet-in-the-middle attacks [BNS19b], to cite only a few. In fact, most of these attacks follow from a re-optimization of classical attacks combined with Grover's search algorithm [Gro96], quantum amplitude amplification (QAA) and amplitude estimation [BHMT02]. These algorithms offer a quadratic speedup ($\mathcal{O}(2^n)$ to $\mathcal{O}(2^{n/2})$), meaning that in general, quantum attacks which only use these tools are limited to an overall quadratic speedup.

In the context of key-recovery attacks on block ciphers, which this paper focuses on, the generic exhaustive search is also quadratically accelerated. As a consequence, the security margin regarding quantum search-based attacks is often equal or higher than classical

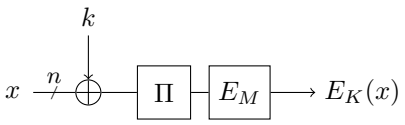

Figure 1: Simplified scenario of FFT-based linear attack against an n-bit block cipher. Here E_M is the part of the cipher with a linear approximation (α, β). Π is a keyless permutation.

attacks (to put it differently, a valid quantum attack using only Grover's search could be "dequantized" into a valid classical attack).

It has been known since the work of Kuwakado and Morii [KM10, KM12] that Simon's algorithm [Sim97] can obtain better speedups than quadratic in symmetric cryptanalysis, leading to total quantum breaks of some symmetric cryptosystems in the *superposition* (Q2) query model. When the amount of data is relatively low with respect to the attack's cost, such attacks can be upgraded to the *classical* (Q1) query model, in which the adversary owns a quantum computer but only accesses classical data [BHN+19]. In this context, some "super-quadratic" speedups are still reachable, using Simon's algorithm as a distinguisher: the offline-Simon algorithm reaches a 2.5 ($\mathcal{O}\left(2^{2.5n}\right)$ to $\mathcal{O}(2^n)$) speedup on the 2-XOR-Cascade block cipher construction [BSS22]. However, Simon's algorithm only distinguishes very specific structures, such as the Even-Mansour cipher [KM12] and reduced-round Feistel networks [KM10, IHM+19, NIDI19, SCQ+23].

Another example of super-quadratic speedup was given in [Hos23], using multidimensional linear (zero-correlation) distinguishers. This highlights the potential for quantum linear attacks.

Quantum Linear Attacks. The first results of quantum linear cryptanalysis were given by Kaplan et al. [KLLN16b], using quantum counting for the linear distinguisher (which estimates the correlation of a given linear approximation), and Grover's search for the key-recovery part.

Advanced linear key-recovery attacks use the Fast Fourier Transform (FFT) to compute quickly the correlations [CSQ07]. In effect, the correlation of the reduced-round cipher for a given subkey guess can be expressed as a convolution of functions. The FFT serves to quickly compute this convolution, allowing to retrieve easily the subkeys which exhibit the largest correlations, suggesting good candidates.

As an example, if we consider the situation of Figure 1, where the cipher has block length n and E_M admits an approximation (α, β) of correlation $|c| \gg 2^{-n/2}$, then k can be retrieved in about $\mathcal{O}(n2^n)$ operations instead of $\mathcal{O}\left(2^n \times c^2\right)$ previously.

This classical key-recovery served in [Sch23] as an inspiration for a quantum one based on the *quantum Fourier transform*. Let $\widehat{\mathrm{cor}}(z)$ be the correlation of $\alpha \cdot x \oplus \beta \cdot E_K(\Pi^{-1}(x) \oplus z)$, whose absolute value is higher when z is equal to k. Then one can compute a so-called *correlation state*:

$$|\mathsf{Cor}\rangle = \sum_z \widehat{\mathrm{cor}}(z)\,|z\rangle \quad .$$

With high probability over the master key K, the good subkey k has amplitude around $|\widehat{\mathrm{cor}}(k)| \simeq |c| \gg 2^{-n/2}$. However, this is not immediately exploitable. The strategy used in [Sch23] was to complete the subkey z into the full cipher key, to check if the full key is good using a few plaintext-ciphertext pairs (like in Grover's search), and perform quantum amplitude amplification (QAA). The correlation state can be seen as an improved starting state for QAA, which creates a speedup with respect to exhaustive search. Unfortunately, very few examples of attacks could be given, and the margin with respect to exhaustive search was limited by a factor $2^{n/2}/c$, which tends to be quite small.

Contribution. In this paper, we extend the framework of [Sch23] and give new quantum linear attacks using the QFT. Our main observation is that *any* distinguisher for E_M allows to turn the correlation state into a *marked* state:

$$\sum_z \widehat{\mathrm{cor}}(z) \left|z\right\rangle \left|(z \stackrel{?}{=} k)\right\rangle \quad .$$

This allows to run QAA, without necessarily guessing the full key. We give examples with two types of distinguishers: using Simon's algorithm, and using a linear approximation.

We apply this idea to the cryptanalysis of Feistel networks in the Q1 and Q2 setting starting with round-reduced LOKI91. Surprisingly, we do not get any benefit from applying a linear approximation. By using Simon's algorithm we reduce the complexity from $2^{33.65}$ (counted in block cipher calls) to $2^{23.73}$ but we need 2^{32} additional classical calls resulting from the round function which is rather problematic for our attacks. In contrast, we provide quantum attacks on reduced-round CAST-128 and CAST-256. These attacks can be seen as accelerations of the previous attacks based on Grover's search and Simon's algorithm, starting from the correlation state to reduce the number of iterations of the key-recovery. On CAST-128, we reduce the complexity from $2^{64.5}$ to $2^{61.1}$. On CAST-256, we increase the number of attacked rounds by 1 compared to the previous Grover-meet-Simon attack, from 23 to 24 [SCQ+23].

Furthermore, we show that when using the same linear approximation as a distinguisher, the QFT-based framework will always improve over the previous attacks of Kaplan et al. [KLLN16b], even though the advantage might become negligible due to the too small correlation. This confirms the potential of QFT-based linear attacks for block cipher cryptanalysis. Using a linear distinguisher, we give another 24-round quantum attack on CAST-256.

Super-quadratic Speedups in Quantum Cryptanalysis. While this is not the case for the block ciphers studied in this paper, it was noticed in [Sch23] that this attack framework may achieve speedups better than quadratic for key-recovery attacks on some specific designs. The example given in [Sch23] is that of an acceleration from $\mathcal{O}\left(2^{2.5n}\right)$ to $\widetilde{\mathcal{O}}(2^n)$, similar to the one offered by the offline-Simon algorithm [BSS22].

Despite reaching seemingly the same gap, our analysis suggests that there is an important difference between these two frameworks. When using the offline-Simon algorithm, the speedup with respect to Grover's search comes entirely from the distinguisher, which internally speeds up from $\mathcal{O}\left(2^{n/2}\right)$ to $\mathcal{O}\left(n^3\right)$, leading to the $0.5n$ advantage in the $2.5n$ speedup. In contrast, part of the speedup that we can gain in the linear attack comes from the key search itself, since we use the correlation statistic to accelerate it. By combining this with an efficient distinguisher, we conjecture that one may reach an even larger speedup than the $2.5n$ – unfortunately, the block cipher would have to be specifically designed for this purpose.

On the use of qRAM / Q2 Queries. The attacks studied in this paper require fast access to large quantum-accessible classical memories (QRACM), and / or superposition (Q2) queries. While both models are quite common in the literature, we favor QRACM over Q2 queries as it is a computational assumption (that memory access is relatively inexpensive) rather than an assumption on the attack scenario (that superposition queries are accessible). How to mitigate this QRACM requirement remains an open question.

Organization. Section 2 gives a collection of results pertaining to the Fourier analysis of Boolean functions, and quantum computing. In Section 3 we recall linear cryptanalysis, key-recovery using the FFT, and the technique of [Sch23]. Next, in Section 4 we introduce

our extension that combines the correlation state with a distinguisher, with a detailed and self-contained analysis.

Our new quantum attacks span the rest of the paper. In Section 5 we use LOKI91 as an example for our different attack strategies. Our new attacks on CAST-128 and CAST-256, using a distinguisher based on Simon's algorithm, are given in Section 6 and Section 7 respectively. Our alternative attack on CAST-256 based on a linear distinguisher is given in Section 8.

Our results on LOKI91, CAST-128 and CAST-256 are summarized respectively in Table 1, Table 2 and Table 3.

2 Preliminaries

In this section, we give some useful definitions related to the Fourier analysis of Boolean functions, and important subroutines of our quantum algorithms.

2.1 Fourier Transforms and Convolutions

In this paper, we will consider discrete Fourier transforms (DFT) defined over $\mathbb{Z}_{2^n}$ and $\mathbb{Z}_2^n$, the latter being called the *Walsh-Hadamard transform* (WHT). We will use the same notation for both, which shall be clear from context. We note $\iota = \sqrt{-1}$ and use $\oplus$ for (bitwise) addition in $\mathbb{Z}_2^n$, not to be confused with $+$ for (modular) addition in $\mathbb{Z}_{2^n}$.

Let $f : \mathbb{Z}_{2^n} \to \mathbb{C}$ be a function, its DFT $\widehat{f} : \mathbb{Z}_{2^n} \to \mathbb{C}$ is defined as:

$$\widehat{f}(x) := \sum_{y \in \mathbb{Z}_{2^n}} \exp(-2\iota\pi xy/2^n) f(y) \ . \tag{1}$$

The *inverse Fourier transform* (IFT) is defined as:

$$\widetilde{f}(x) := \sum_{y \in \mathbb{Z}_{2^n}} \exp(2\iota\pi xy/2^n) f(y) \ . \tag{2}$$

We have the following property:

$$\widetilde{\widehat{f}} = \widehat{\widetilde{f}} = 2^n f \ . \tag{3}$$

The DFT can be generalized to functions with multiple input variables. More precisely, if f is defined over a product $\prod_{0 \leq i \leq m-1} \mathbb{Z}_{2^{n_i}}$ (where $m, n_0, \ldots, n_{m-1} \in \mathbb{N}$), the DFT is defined as:

$$\widehat{f}(x_0, \ldots, x_{m-1}) = \sum_{y_0, \ldots, y_{m-1}} \prod_i \exp(-2\iota\pi x_i y_i / 2^{n_i}) f(y_0, \ldots, y_{m-1}) \ , \tag{4}$$

and the IFT similarly. In particular, for the special case that $m = n$ and $n_0 = \cdots = n_{m-1} = 1$, the DFT of a function f defined over $\mathbb{Z}_2^n$ is called the *Walsh-Hadamard Transform* (WHT)[1] and $\widehat{f} : \mathbb{Z}_2^n \to \mathbb{C}$ is defined as:

$$\widehat{f}(x) = \sum_{y \in \mathbb{Z}_2^n} (-1)^{x \cdot y} f(y) \ . \tag{5}$$

In this case, the transformation is an involution (the WHT and its inverse are the same), so we use only the notation $\widehat{f}$.

[1] Some authors make a difference between the DFT and the WHT, depending on the normalization coefficient.

There is a natural bridge between the DFT and the so-called *discrete convolution*, which, for two functions f, g defined over $\prod_{0\le i\le m-1}\mathbb{Z}_{2^{n_i}}$, is given by:

$$(f\star g)(x_0,\ldots,x_{m-1}) = \sum_{y_0,\ldots,y_{m-1}} f(y_0,\ldots,y_{m-1})g(x_0-y_0,\ldots,x_{m-1}-y_{m-1}) \ , \quad (6)$$

and in the case of $\mathbb{Z}_2^n$, is defined by:

$$(f\star g)(x) = \sum_y f(y)g(x\oplus y) \ . \quad (7)$$

The connection between the DFT and the discrete convolution is given through the following identity, which we will used througout this paper.

Lemma 1 (Convolution theorem)**.** *For two functions f, g defined over $\prod_{1\le i\le \ell}\mathbb{Z}_{2^{n_i}}$:*

$$\left(\prod_{i=1}^{\ell} 2^{n_i}\right)(f\star g) = \widetilde{\widehat{f}\cdot\widehat{g}} \ . \quad (8)$$

Moreover, we will also use the following well-known fundamental properties.

Lemma 2 (Product)**.** *For a family of functions $\{f_i : \mathbb{Z}_{2^{n_i}} \to \mathbb{C}\}_{1\le i\le \ell}$, let $f_1 f_2\cdots f_\ell : \prod_{1\le i\le \ell}\mathbb{Z}_{2^{n_i}} \to \mathbb{C}$ denote the function defined by $(f_1 f_2\cdots f_\ell)(x_1,\ldots,x_\ell) = \prod_{1\le i\le \ell} f_i(x_i)$. Then $\widehat{f_1 f_2\cdots f_\ell} = \widehat{f}_1\widehat{f}_2\cdots\widehat{f}_\ell$ and $\widetilde{f_1 f_2\cdots f_\ell} = \widetilde{f}_1\widetilde{f}_2\cdots\widetilde{f}_\ell$.*

Lemma 3 (Parseval identity)**.** *For a function f defined over $\prod_{1\le i\le \ell}\mathbb{Z}_{2^{n_i}}$:*

$$\sum_y \widehat{f}^2(y) = \sum_y \widetilde{f}^2(y) = \left(\prod_{i=1}^{\ell} 2^{n_i}\right)\sum_y f^2(y) \ . \quad (9)$$

In particular, if f is a function to $\{-1,1\}$:

$$\sum_y \widehat{f}^2(y) = \sum_y \widetilde{f}^2(y) = \left(\prod_{i=1}^{\ell} 2^{n_i}\right)^2 \ . \quad (10)$$

2.2 Preliminaries of Quantum Computing

We assume some familiarity of the reader with the basics of quantum computing [NC02] such as quantum states, qubits, basic operations and the quantum circuit model. For a unitary U implemented as a quantum circuit, its inverse $U^\dagger$ can be implemented by reverting the sequence of operations of U. We denote the gate count of a quantum algorithm $\mathcal{A}$ as $\mathcal{G}(\mathcal{A})$.

In this paper, we are interested in quantum key-recovery attacks on block ciphers. These attacks are quantum algorithms described as quantum circuits. For any block cipher E_K of key length $|K|$, there is a quantum exhaustive search attack of complexity $\mathcal{O}\left(2^{|K|/2}\right)$ using negligible memory and data, by Grover's quantum search algorithm [Gro96]. Similarly to classical cryptanalysis, our goal is to obtain a smaller time complexity.

Q1 and Q2 Settings. In symmetric quantum cryptanalysis, there is a gap between the *Q1 setting* (only classical chosen-plaintext queries available) and the *Q2 setting*, where the adversary has access to a quantum black-box oracle for the cipher (and sometimes its inverse): $|x\rangle|0\rangle \mapsto |x\rangle|E_K(x)\rangle$. It is known that some ciphers are secure in Q1 but insecure in Q2, like the Even-Mansour cipher [KM12, ABKM22]. Yet, as we explain below, if the key length is twice as large as the block size, the Q2 setting can be replaced by a hardware assumption (the QRACM model) at the expense of a large classical data complexity.

Quantum RAM. The *quantum RAM* model is an assumption used commonly in quantum algorithms to minimize their gate count. Quantum RAM allows complex memory operations to be made in quantum circuits, provided that the so-called "qRAM gate" has cost 1 (like other basic gates). In this paper, we will rely only on *quantum random-access classical memory* (QRACM). This is a large classical memory array M which can be accessed through qRAM gates of the form:

$$|i\rangle |y\rangle \xmapsto{\mathsf{qRAM}} |i\rangle |y \oplus M_i\rangle \tag{11}$$

where M_i is the bit (or the data) at index i in the array. We assume that such qRAM gates can be efficiently implemented.

Remark 1 (Relation between QRACM and the Q2 model). Consider an n-bit block cipher with $|K|$-bit keys, where $|K| \geq 2n$. Then the complexity of Grover's exhaustive search is a bit larger than 2^n. This means that in an attack, we can afford to query the entire classical codebook of the cipher and store it in a QRACM of size 2^n. After doing so, the Q2 oracle to the cipher (and its inverse) can be implemented by querying the QRACM.

Complexity Estimates. The cost of a quantum circuit is best estimated by its number of qubits, gates, and its total depth. In this paper, we will favor a simpler but more imprecise estimation of the complexity.

First, regarding the memory complexity: as the number of computational qubits used in our attacks will remain relatively small (polynomial in the block size), we ignore it. We focus instead on the amount of QRACM used, counted in blocks.

Second, our algorithms are sequential, like Grover's search, so instead of comparing the depth we focus on the gate count. We approximate it by counting in calls to the attacked block cipher, i.e., relatively to the cost of an implementation of the cipher as a quantum circuit. A Q2 query corresponds to one call. Following the relation between QRACM and Q2 queries, we assume that a QRACM query has equivalent cost. Whenever we need to implement the cipher's sub-components, we may upper bound the cost by 1 call. Other components of the algorithms (such as Grover's diffusion transform) can typically be neglected.

2.3 Simon's Algorithm

Simon's algorithm [Sim97] is a polynomial-time quantum algorithm to solve the following problem.

Problem 1 (Hidden Boolean period)**.** *Given a quantum oracle O_f for a two-to-one function $f : \{0,1\}^n \to \{0,1\}^m$ such that $\exists s, \forall x, y, f(x) = f(y) \iff y = x \vee y = x \oplus s$, find s.*

This problem has been particularly useful in symmetric cryptanalysis, leading to both Q2 [KM10, KM12, KLLN16a] and Q1 [BHN+19, BSS22] attacks. In these applications, however, the function is not two-to-one. Rather, being built from (supposedly) secure cryptographic components, it behaves statistically like a random function. It is well-known that the original algorithm from Simon still functions in such a case [KLLN16a].

Another restriction is when the function has a small codomain, for example a single-bit output. Notably, May and Schlieper considered the case of a compressed function [MS22]. Obviously, a single-bit output subsumes the case of any function f, which can either be compressed or truncated using the method of [HS18, BBC+21].

While most works use Simon's algorithm as is, the algorithm that we give below moves slightly away from it, as its goal is to be tailored to the single-bit output case.

Simon's Algorithm as a Fourier Analysis Problem. In the case $m = 1$, consider the quantum subroutine of Algorithm 1, which calls a *phase* oracle for f: $|x\rangle \mapsto (-1)^{f(x)} |x\rangle$. Such an oracle can easily be constructed from any implementation of f.

Algorithm 1 Simon's algorithm with a phase oracle

1: Initialize a register for x: $|0\rangle$
2: Apply a Hadamard transform: $\frac{1}{2^{n/2}} \sum_x |x\rangle$
3: Call f: $\frac{1}{2^{n/2}} \sum_x (-1)^{f(x)} |x\rangle$
4: Apply a Hadamard transform: $\frac{1}{2^n} \sum_y \left(\sum_x (-1)^{x \cdot y} (-1)^{f(x)} \right) |y\rangle$

Here $\widehat{g}(y) := \sum_x (-1)^{x \cdot y} (-1)^{f(x)}$ is a Walsh-Hadamard coefficient of the function $g : x \mapsto (-1)^{f(x)}$, and this subroutine merely creates a quantum state whose amplitudes are these coefficients (rescaled): $\frac{1}{2^n} \sum_y \widehat{g}(y) |y\rangle$.

The following property of the coefficients is at the heart of Simon's algorithm.

Lemma 4. *If f is periodic of period $s \neq 0$ (for all x, $f(x \oplus s) = f(x)$), then for all y such that $y \cdot s = 1$, $\widehat{g}(y) = 0$.*

Proof. We first partition the input space $\{0,1\}^n$ into two subsets X and $X \oplus s$. Then, since f is periodic, we have:

$$\widehat{g}(y) = \sum_{x \in X} (-1)^{x \cdot y} (-1)^{f(x)} + (-1)^{(x \oplus s) \cdot y} (-1)^{f(x \oplus s)} = \sum_{x \in X} (-1)^{x \cdot y} (1 + (-1)^{y \cdot s}) (-1)^{f(x)} = 0 \ .$$

□

The other Walsh-Hadamard coefficients are equal to $2 \sum_{x \in X} (-1)^{x \cdot y} (-1)^{f(x)}$. In particular, the probability to measure 0 in the output state is equal to: $\frac{|\widehat{g}(0)|^2}{2^{2n}} = |\#\{x, f(x) = 0\} - \#\{x, f(x) = 1\}|^2 / 2^{2n} = \mathcal{O}(1/2^n)$ if the function is random. The exact probability to measure each y depends on the function f, but if it behaves like a random function, we can expect to measure each y with roughly the same probability. These considerations are studied extensively in [Bon21], and we rely on their analysis for concrete estimates.

Therefore, after running Algorithm 1 $n+k$ times (for some well-chosen k) and performing a measurement after each run, we obtain a set of vectors $y_1, \ldots, y_{n+k}$ that define a linear system on s, from which we can obtain s. If the function is random and not periodic, then we will simply measure random vectors and the family $y_1, \ldots, y_{n+k}$ will be full rank.

This observation has the following by-product: We can use Algorithm 1 to *detect*, whether the function is periodic or whether it is random (and hence not periodic). In other words: We can use this algorithm to *distinguish* periodic functions from random functions. In this paper we will apply Algorithm 1 with this objective.

Running the Procedure Coherently. Computing the rank of the family can be performed coherently, giving us a function to test whether f is periodic or not. This test can make errors: even in the random case, we can happen to pick random vectors $y_1, \ldots, y_{n+k}$ that do not form a full-rank family. Increasing k reduces such errors exponentially.

In general, we use Simon's algorithm inside a Quantum Amplitude Amplification routine. Following Bonnetain's analysis of the offline-Simon algorithm (Heuristic 5 in [Bon21]), we consider that if Simon's algorithm is embedded in a search with $2^{\ell/2}$ iterations, $n + \ell + \alpha + 4$ vectors are sufficient to succeed with an overall probability of at least $1 - 2^{-\alpha}$.

2.4 Quantum Distinguishers on Feistel Ciphers

Kuwakado and Morii [KM10] showed that the 3-round Feistel could be distinguished from a random permutation in polynomial time using Simon's algorithm. Since then,

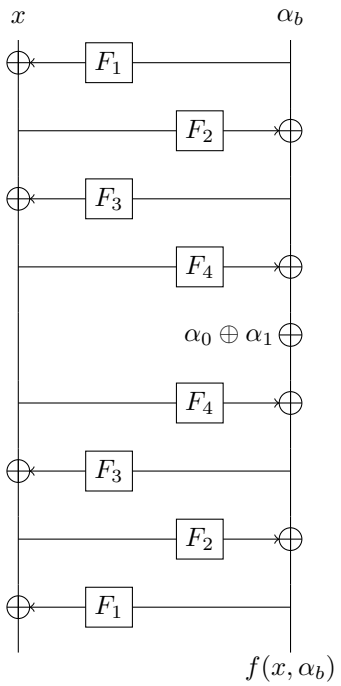

Figure 2: 4-round Feistel distinguisher of [IHM+19].

many quantum distinguishers on (Generalized) Feistel networks have been given [IHM+19, DLW19, NIDI19, BNS19a, HK20, CHLS20, SCQ+23, CLS22, XWY+24]. In this paper, we will use a distinguisher on the 4-round Feistel of [IHM+19] and a distinguisher on the 17-round CAST-256 structure of [SCQ+23]. We recall the former in Figure 2. Being generic, we will apply it to both LOKI91 and CAST-128.

Let α_0, α_1 be a pair of constants and let E be the 4-round Feistel with unknown round functions F_1, F_2, F_3, F_4. Let $f(x, \alpha_b)$ be constructed as in Figure 2, by calling E on input (x, α_b), swapping the outputs, XORing $\alpha_0 \oplus \alpha_1$, calling E^{-1} and truncating to a single branch. Then:

$$f(x, \alpha_0) = f(x \oplus F_1(\alpha_0) \oplus F_1(\alpha_1), \alpha_1) \oplus (\alpha_0 \oplus \alpha_1) \ . \tag{12}$$

Indeed, if we introduce a difference $\alpha_0 \oplus \alpha_1$ in the constant input branch, we only need to correct the value of x by $(F_1(\alpha_0) \oplus F_1(\alpha_1))$ and we get the same output, up to a difference $\alpha_0 \oplus \alpha_1$. This means that the function:

$$F(x, b) = f(x, \alpha_b) \oplus \alpha_b \tag{13}$$

is periodic of period $(F_1(\alpha_0) \oplus F_1(\alpha_1), 1)$. Using Simon's algorithm on this function allows to distinguish E from a random permutation. Furthermore each call to F contains one call to E and one call to E^{-1}.

Extension to a Grover-meet-Simon Distinguisher. In the applications of this paper, we will typically extend a Simon distinguisher by guessing the keys of additional rounds, turning it in practice into a Grover-meet-Simon [LM17] distinguisher. Given access to the cipher, one performs a Grover's search over the additional round keys, expecting one solution. A value for them is tested by running the Simon distinguisher on the reduced-round cipher (to which we have now access since the outer keys are guessed). In the random case, no key leads to a periodic function. Otherwise, the right key (that leads to a periodic function) is found by Grover's search.

2.5 Quantum Amplitude Amplification (QAA)

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, U be a unitary operator acting on n-qubit quantum states, and p be the probability that the outcome $x \in \{0,1\}^n$ satisfies $f(x) = 1$ when the quantum state $U|0^n\rangle$ is measured. In cryptanalysis, the function f is typically defined so that $f(x) = 1$ iff x matches a secret value that we want to recover with a high probability. The Quantum Amplitude Amplification (QAA) is an algorithm that amplifies the probability p by iteratively applying U and the (phase) oracle of f.

More precisely, define the unitary operator:

$$Q(U, f) := U\mathcal{S}_0 U^\dagger \mathcal{S}_f \,,$$

where the unitary operators $\mathcal{S}_f$ and $\mathcal{S}_0$ are defined as:

$$\mathcal{S}_f |x\rangle = (-1)^{f(x)} |x\rangle \,,$$
$$\mathcal{S}_0 |x\rangle = (-1)^{x \overset{?}{=} 0} |x\rangle \,.$$

The operator $\mathcal{S}_f$ is a phase oracle, flipping the sign of the state $|x\rangle$ only when $f(x) = 1$. The operator $\mathcal{S}_0$ flips the phase only of the $|0^n\rangle$ state, hence acting as a reflection about the subspace orthogonal to $|0^n\rangle$.

The composition $U\mathcal{S}_0 U^\dagger$ represents the conjugation of $\mathcal{S}_0$ by U, which geometrically corresponds to a reflection about the state $U|0^n\rangle$.

Thus, $Q(U, f)$ performs a composite reflection:

1. $\mathcal{S}_f$ reflects about the *good* subspace, namely the solutions to $f(x) = 1$.
2. $U\mathcal{S}_0 U^\dagger$ reflects about the initial state $U|0^n\rangle$.

This double reflection defines a rotation in the 2D subspace spanned by the *good* ($f(x) = 1$) and *bad* ($f(x) = 0$) components of $U|0^n\rangle$, thereby increasing p.

Given this, the following theorem holds.

Theorem 1 ([BHMT02])**.** *Let $m := \lfloor \pi/(4 \arcsin \sqrt{p}) \rfloor$. If we measure the quantum state $(Q(U, f))^m U|0^n\rangle$, then the outcome $x \in \{0,1\}^n$ satisfies $f(x) = 1$ with a probability of at least $\max\{p, 1-p\}$. In other words, we can amplify the probability of measuring such x to $\Theta(1)$ by applying U, $U^\dagger$, $\mathcal{S}_0$ and $\mathcal{S}_f$ for $\mathcal{O}\left(\sqrt{1/p}\right)$ times.*

This matches Grover's search [Gro96] when U is the Hadamard transform.

Reaching a probability of success close to 1 requires knowing the parameter p. However, if only a lower bound on p is given, running a QAA with a random number of iterates still allows to succeed with good probability.

Lemma 5 (QAA with unknown success probability [SS24, Lemma 2])**.** *Assume that $p \geq p_{\min}$. There is a quantum procedure that finds x such that $f(x) = 1$ with probability greater than 0.5 using on average $2\left\lceil \frac{1.21}{\sqrt{p_{\min}}} \right\rceil + 2$ calls to U and $U^\dagger$.*

2.6 Quantum Counting (QC)

Again, let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let $X := \#\{x : f(x) = 1\}$. The goal of quantum counting (QC) is to estimate X. We will make use of the following theorem:

Theorem 2 ([BHMT02, Theorem 12])**.** *There is a quantum algorithm* `Est_Amp` *estimating the amplitude* $a \in \{0,1\}$ *of measuring a "good" outcome, namely an* x *such that* $f(x) = 1$*. More precisely, for any positive integers* k *and* q*,* `Est_Amp` *outputs* $\tilde{a}$ *with* $(0 \leq \tilde{a} \leq 1)$ *such that*

$$|\tilde{a} - a| \leq 2\pi k \sqrt{\frac{a(1-a)}{q^2}} + k^2 \left(\frac{\pi}{q}\right)^2$$

with probability at least $\frac{8}{\pi^2}$ *when* $k = 1$ *and with probability greater than* $1 - \frac{1}{2(k-1)}$ *for* $k \geq 2$*. It uses exactly* q *evaluations of* f*. If* $a = 0$ *then* $\tilde{a} = 0$ *with certainty, and if* $a = 1$ *and* q *is even, then* $\tilde{a} = 1$ *with certainty.*

Suppose that a quantum circuit implementing f with gate count $G(f)$ is availabe. Assign $k = 1$. Let $a = \frac{X}{2^n}$. Based on Theorem 2, we can observe that, for any positive integer q, there is a unitary operator QC_q satisfying the following properties:

1. Measuring (some part of) the quantum state $\mathsf{QC}_q |0\rangle$, the outcome $\hat{X}$ is an integer such that $\left|\hat{X} - X\right| \leq 2\pi\sqrt{X(2^n - X)/q^2} + (\pi \cdot 2^{n/2}/q)^2$ with a probability of at least $\frac{8}{\pi^2} \approx 0.8$.
2. QC_q can be implemented on a quantum circuit such that it makes exactly q queries to f and its gate count is approximately $q \cdot G(f)$.

Following [KLLN16b], we will use the algorithm for linear distinguishers.

3 Quantum Linear Cryptanalysis using the QFT

In this section, we recollect previous work on linear key-recovery attacks using the FFT, and the QFT in the quantum setting. First, we recall the concept of linear cryptanalysis, the *experimental correlation* statistic and the (classical) FFT technique for computing correlations. Second, we present the technique of [Sch23], the computation of the *correlation state* using the QFT and how it extends to a key-recovery attack.

3.1 Linear Cryptanalysis

Linear cryptanalysis, introduced by Matsui [Mat93, Mat94], is a powerful cryptanalysis method based on linear distinguishers.

Let E_K be an n-bit block cipher. A *linear approximation* on a reduced-round version E_M is given by a pair of *masks* (α, β) such that the *correlation*:

$$\mathrm{cor}_K(\alpha, \beta) := \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\alpha \cdot x \oplus \beta \cdot E_M(x)} \tag{14}$$

is much larger than for a random function, i.e., much larger than $2^{-n/2}$. In general, E_M is a keyed function, and the correlation depends on the key. The strength of the approximation is determined by its ELP (*Expected Linear Potential*):

$$\mathrm{ELP} := \frac{1}{2^{|K|}} \sum_{K \in \mathbb{F}_2^{|K|}} \mathrm{cor}_K(\alpha, \beta)^2 \ . \tag{15}$$

The scenario that we consider is represented in Figure 3. We assume having access to a database $\mathcal{D}$ containing $N := |\mathcal{D}|$ distinct known plaintext-ciphertext pairs. After guessing the subkey material $k^{\mathrm{in}}, k^{\mathrm{out}}$ as $z^{\mathrm{in}}, z^{\mathrm{out}}$, one computes the *experimental correlation*:

$$\widehat{\mathrm{cor}}(z^{\mathrm{in}}, z^{\mathrm{out}}) = \frac{1}{N} \sum_{(x, E_K(x)) \in \mathcal{D}} (-1)^{\alpha \cdot F^L_{z^{\mathrm{in}}}(x \oplus z^{\mathrm{out}})} (-1)^{\beta \cdot F^R_{z^{\mathrm{in}}}(E_K(x))} \ . \tag{16}$$

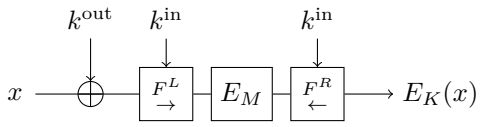

Figure 3: Attack scenario (figure adapted from [Sch23]). Here E_M is the part of the cipher with a linear approximation (α, β), F_L and F_R are functions which allow to compute the value in the input and output masks respectively.

When $z^{\text{in}}, z^{\text{out}} \neq k^{\text{in}}, k^{\text{out}}$, the experimental correlation is the correlation of (α, β) for a random-looking permutation. Its statistic is given by the following *wrong-key randomization hypothesis*.

Assumption 1 (Wrong-key randomization)**.** *At fixed K, $\widehat{\text{cor}}(z)$ is a random variable over z following a normal distribution $\mathcal{N}(0, \sigma_W^2)$ where $\sigma_W^2 = \left(\frac{2^n - N}{2^n - 1}\right)\frac{1}{N} + 2^{-n}$.*

When $z^{\text{in}}, z^{\text{out}} = k^{\text{in}}, k^{\text{out}}$, the experimental correlation becomes the correlation of E_M. Its statistic is given by the *right-key randomization hypothesis*. This hypothesis differs whether the linear approximation has many dominating characteristics, whose correlations interfere, or a single one. In this paper, we use linear approximations with a single characteristic of correlation $\pm c$ (where the sign depends on the master key). We will make the assumption that this characteristic dominates. On the contrary in [Sch23] the cases studied had many characteristics, and the statistic is different [BN16].

Assumption 2 (Right-key randomization, single characteristic, see Theorem 4 in [BN17])**.** *For the right subkey guess $(k^{\text{in}}, k^{\text{out}})$, $\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})$ is a random variable over $(k^{\text{in}}, k^{\text{out}})$ following a normal distribution $\mathcal{N}(\mu_R, \sigma_R^2)$ where $\mu_R = \pm c$ and $\sigma_R^2 = \frac{1}{N} + \text{ELP} - c^2$*

We make furthermore the simplifying assumption that $\text{ELP} = c^2$. Classical cryptanalysis distinguishes these two cases thanks to their different statistics. Quantum cryptanalysis relies mostly on a *lower bound* for the correlation of the right key. Indeed, as a consequence of right-key randomization, with probability 0.95, a random key will satisfy:

$$|\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}}) - (\pm c)| \leq \frac{2}{\sqrt{N}} \implies c + \frac{2}{\sqrt{N}} \geq |\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})| \geq c - \frac{2}{\sqrt{N}} \ . \tag{17}$$

3.2 The FFT Technique

Collard et al. [CSQ07] introduced an efficient method of evaluating the experimental correlations when part of the key is XORed to the internal state of the cipher. In Figure 3, we will now separate the roles of k^{in} and k^{out}, and rewrite the experimental correlation (Equation 16) as a convolution of two functions over $\mathbb{F}_2^n$:

$$\begin{cases} f_{z^{\text{in}}}(x) := (-1)^{\alpha \cdot F^L_{z^{\text{in}}}(x)} \\ g_{z^{\text{in}}}(x) := \mathbf{1}[x \in \mathcal{D}](-1)^{\beta \cdot F^R_{z^{\text{in}}}(E_K(x))} \\ \widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}}) = \frac{1}{N}\left(f_{z^{\text{in}}} \star g_{z^{\text{in}}}\right)(z^{\text{out}}) \ . \end{cases} \tag{18}$$

Here $\mathbf{1}[x \in \mathcal{D}]$ is 1 if x appears in the database of known plaintexts, and 0 otherwise.

Then, in order to compute efficiently the correlation, one first guesses z^{in}, then evaluates all $\widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}})$ using the convolution theorem (Lemma 1). The Walsh-Hadamard transforms of $f_{z^{\text{in}}}$ and $g_{z^{\text{in}}}$ are evaluated in time $\mathcal{O}(n2^n)$ by the Fast Walsh-Hadamard transform algorithm (a special case of the more general Fast Fourier Transform), then multiplied pointwise, and one takes the WHT again to obtain the correlations for all z^{out}. This method requires $\mathcal{O}(2^{|k^{\text{in}}|} \times n2^n)$ time and $\mathcal{O}(2^n)$ space.

More generally as shown in [FN20], one can separate the k^{out} part in the first and last rounds of the cipher. This requires a *distillation* phase of the data which reorders the database by separating the input and output differently, since some part of $E_K(x)$ will now go in f and some part of x will go in g.

Finally, [WWBC14] introduced a variant of this FFT-based approach in which parts of the key k^{out} can be added with $\oplus$, and parts with modular additions. In fact, this only changes the input space of the functions $f_{z^{\text{in}}}$ and $g_{z^{\text{in}}}$ (where they still have the same definition); one uses the more general convolution theorem (Lemma 1) to express the correlation.

3.3 Quantum Version: Correlation State

A crucial step in the QFT-based linear cryptanalysis introduced in [Sch23]is the computation of a *correlation state*. This state is given by the superposition over all key candidates with the corresponding amplitudes in this superposition being determined by the respective experimental correlation. In this way, we obtain a quantum state, where the information about the linear approximation is directly encoded inside the amplitudes (i.e. in the measurement probabilities). In particular, as a consequence, the right key has the largest amplitude among all key candidates as it has the experimental correlation with the largest modulus.

In mathematical terms, the correlation state, in the way we use it here in this paper, is defined as follows:

$$|\mathsf{Cor}_{z^{\text{in}}}\rangle := \frac{\sqrt{N}}{2^{n/2}} \sum_{z^{\text{out}}} \widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}}) \, |z^{\text{out}}\rangle \tag{19}$$

where the sum is on all 2^n possibilities of z^{out}.

Remark 2 (Normalization of $|\mathsf{Cor}_{z^{\text{in}}}\rangle$). As $|\mathsf{Cor}_{z^{\text{in}}}\rangle$ is a quantum state, it has to be normalized. This is the reason why its definition (19) includes the prefactor $\frac{\sqrt{N}}{2^{n/2}}$. Indeed, following the wrong-key randomization hypothesis, $\widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}})$ follows a normal distribution with average 0 and variance $\sigma^2 = \frac{2^n - N}{(2^n-1)N} + 2^{-n} \simeq \frac{1}{N} + 2^{-n}$, meaning that $\frac{1}{\sigma^2}\widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}})^2$ follows a χ^2 distribution with average 1. By summing over all wrong keys, we have:

$$\sum_{z^{\text{out}}} \widehat{\text{cor}}(z^{\text{in}}, z^{\text{out}})^2 \simeq 2^n \sigma^2 \simeq 2^n \left(2^{-n} + \frac{2^n - N}{(2^n-1)N}\right) \simeq 1 + \frac{2^n - N}{N} = \frac{2^n}{N} \ . \tag{20}$$

We can neglect the correlation of the right key, as in practical applications it does not dominate (unless the function is very close to linear). Notice that if the full codebook is available, $N = 2^n$ and the normalization factor of Equation 19 is simplified into 1.

This version differs significantly from the classical one. First, in the way the correlation state is constructed, which we explain below. Second, in the way it can be used for cryptanalysis. Indeed, while classical linear cryptanalysis uses very small values of the correlations to distinguish from the random case, the same cannot be done generically in the quantum setting. Given an algorithm that produces $|\mathsf{Cor}_{z^{\text{in}}}\rangle$, finding the value of z^{out} which has the largest amplitude in the state seems to be a very difficult problem.

In order to avoid this, the idea of [Sch23] is to guess the remainder of the key, which "completes" the correlation state into a superposition over all possible key guesses, where the amplitude on the good subkey is larger (due to the larger correlation that one has at the beginning).

Indeed, by Equation 17, we know that $|\widehat{\text{cor}}(k^{\text{in}}, k^{\text{out}})| \geq \sqrt{\text{ELP}} - \frac{2}{\sqrt{N}}$. Thus in $|\mathsf{Cor}_{k^{\text{in}}}\rangle$ the absolute amplitude on k^{out} is lower bounded by:

$$\frac{\sqrt{N}}{2^{n/2}} \left(\sqrt{\text{ELP}} - \frac{2}{\sqrt{N}}\right) = \frac{\sqrt{N}}{2^{n/2}} \sqrt{\text{ELP}} - 2^{-n/2+1} \ . \tag{21}$$

Remark 3 (On the normalization). The amount of data intervenes as a scaling factor in this amplitude, which directly impacts the time complexity. This is why we will most often consider having access to the full codebook.

3.4 Construction of the Correlation State

Since in the end, our goal is only to bound the amplitude on the correct subkey, we only need a subroutine that on input $|k^{\text{in}}\rangle|0\rangle$, returns $|k^{\text{in}}\rangle|\mathsf{Cor}_{k^{\text{in}}}\rangle$. We do not need this subroutine to work for all possible z^{in}. In practice, this simply means that we can afford relatively lower probabilities of success.

We slightly adapt the subroutine given in [Sch23].

Lemma 6 (From [Sch23], adapted)**.** *Assume that we have a pair of unitary operations to access $\mathcal{D}$:*

$$\begin{cases} |0\rangle \xmapsto{\mathsf{Init}_{\mathcal{D}}} \frac{1}{\sqrt{N}} \sum_{x\in\mathcal{D}} |x\rangle \\ |x\rangle|0\rangle \xmapsto{\mathsf{Query}_{\mathcal{D}}} |x\rangle|\mathcal{D}(x)\rangle \end{cases} \tag{22}$$

where $\mathcal{D}(x) := E_K(x)$ if $x \in \mathcal{D}$ and $\perp$ otherwise.

Let fFourier *be a unitary that maps:*

$$|k^{\text{in}}\rangle|x\rangle|0\rangle \xmapsto{\mathsf{fFourier}} |k^{\text{in}}\rangle|x\rangle\left(\frac{\widehat{f}_{k^{\text{in}}}(x)}{G}|0\rangle + |*\rangle\right) \quad , \tag{23}$$

where $|*\rangle$ is a superposition of non-zero basis states, and G is an upper bound on the moduli of all Fourier coefficients of $\widehat{f}_{k^{\text{in}}}$ (we make no assumption on the behaviour of the unitary for $z^{\text{in}} \neq k^{\text{in}}$).

Then the unitary described in Algorithm 2 maps (up to negligible error):

$$|k^{\text{in}}\rangle|0\rangle \xmapsto{\mathsf{Corcomp}} |k^{\text{in}}\rangle|\mathsf{Cor}_{k^{\text{in}}}\rangle \quad . \tag{24}$$

Its gate count can be approximated by:

$$\left(\frac{\pi}{2}\frac{G}{2^{n/2}} + 3\right)\left(2\mathcal{G}(\mathsf{Query}_{\mathcal{D}}) + \mathcal{G}(\mathsf{Init}_{\mathcal{D}}) + 2\mathcal{G}(\mathsf{F}^{\mathsf{R}}) + \mathcal{G}(\mathsf{fFourier}) + \mathcal{O}(n)\right) \quad . \tag{25}$$

Proof. The only substantial technicality in Algorithm 2 is the number of iterations to perform at step 6. We notice that the total amplitude on 0 at this point is:

$$\frac{1}{G^2 2^n N}\sum_x (\widehat{f}_{k^{\text{in}}}(x))^2(\widehat{g}_{k^{\text{in}}}(x))^2$$

By Parseval's equality and the convolution theorem (see Subsection 2.1) we have:

$$\sum_x (\widehat{f}_{k^{\text{in}}}(x))^2(\widehat{g}_{k^{\text{in}}}(x))^2 = 2^n \sum_x \left(f_{k^{\text{in}}} \star g_{k^{\text{in}}}\right)^2 = 2^n N^2 \sum_x \widehat{\text{cor}}(z^{\text{in}}, x)^2 \quad .$$

Using the approximation $\sum_x \widehat{\text{cor}}(z^{\text{in}}, x)^2 \simeq \frac{2^n}{N}$ from Equation 20, the total amplitude on 0 is equal to:

$$\frac{1}{G^2 2^n N} \times 2^n N^2 \times \frac{2^n}{N} = \frac{2^n}{G^2} \quad ,$$

meaning that (roughly) $\frac{\pi}{2}\frac{G}{2^{n/2}}$ QAA iterates are required.

□

Algorithm 2 Implementation of Corcomp, adapted from [Sch23]

1: Call $\mathsf{Init}_{\mathcal{D}}$: $|k^{\text{in}}\rangle \frac{1}{\sqrt{N}} \sum_{x\in\mathcal{D}} |x\rangle$

2: Call $\mathsf{Query}_{\mathcal{D}}$: $|k^{\text{in}}\rangle \frac{1}{\sqrt{N}} \sum_{x\in\mathcal{D}} |x\rangle |E(x)\rangle |F^R(E(x))\rangle$

3: Compute $(-1)^{\beta\cdot F^R(E(x))}$ by a controlled phase flip, erase F^R and E:
$|k^{\text{in}}\rangle \frac{1}{\sqrt{N}} \sum_x g_{k^{\text{in}}}(x) |x\rangle$

4: Compute a Hadamard transform $|k^{\text{in}}\rangle \frac{1}{\sqrt{N}2^{n/2}} \sum_x \widehat{g}_{k^{\text{in}}}(x) |x\rangle$

5: Compute fFourier $|k^{\text{in}}\rangle \frac{1}{\sqrt{N}2^{n/2}} \sum_x \widehat{g}_{k^{\text{in}}}(x) |x\rangle \left(\frac{\widehat{f}_{k^{\text{in}}}(x)}{G} |0\rangle + |*\rangle\right)$

6: Amplify the part of the state which ends with 0. This requires to repeat all the previous operations. We obtain the state:

$$|k^{\text{in}}\rangle \frac{1}{2^n\sqrt{N}} \sum_x \widehat{g}(x)\widehat{f}(x) |x\rangle \quad .$$

7: Compute a Hadamard transform of the output. By the convolution theorem we obtain:

$$|k^{\text{in}}\rangle \frac{1}{2^{n/2}\sqrt{N}} \sum_x (f \star g) |x\rangle = |\mathsf{Cor}_{k^{\text{in}}}\rangle \quad .$$

Remark 4. Note that we always have $G \leq 2^n$. However, as the gate count in Lemma 6 scales in G, it is important to find a better bound on G. This is one of the tasks in our concrete attacks in the Sections 5, 6, 7 and 8.

Since we can only estimate (rather than compute exactly) the number of QAA iterates to perform in Corcomp, we expect a minor relative error. This error simply modifies the probability to measure the right $k^{\text{in}}, k^{\text{out}}$ in the obtained state, so it does not disrupt the QAAs that we perform later, and remains inconsequential for our attacks.

For a random function, the bound G is such that $G = \widetilde{\mathcal{O}}(2^{n/2})$, ensuring that the number of QAA iterates is polynomial in n. However, the function f does not always behave as random. Furthermore, one needs to implement the fFourier unitary, which is a technical step.

The unitary fFourier is a special case of *quantum state preparation*. We recall the method of [SLSB19], which is used in [Sch23].

Lemma 7 (Amplitude transduction by comparison, from [SLSB19])**.** *Given access to a unitary:* $|k^{\text{in}}\rangle |x\rangle |0\rangle \mapsto |k^{\text{in}}\rangle |x\rangle |\widehat{f}_{k^{\text{in}}}(x)\rangle$ *which computes* $\widehat{f}_{k^{\text{in}}}(x)$ digitally*, there exists an implementation of* fFourier *(which computes* $\widehat{f}_{k^{\text{in}}}(x)$*, rescaled by* G*, in the amplitudes) that calls this unitary and its inverse once and uses* $\mathcal{O}(n)$ *additional gates.*

Proof. In the following we write $\widehat{f}_{k^{\text{in}}}(x) = m(k^{\text{in}}, x)\exp(\iota\theta(k^{\text{in}}, x))$, where $m(k^{\text{in}}, x)$ is a positive real, and assume (for simplicity) that we can use some fixed-precision encoding of $m(k^{\text{in}}, x)$ and $\theta(k^{\text{in}}, x)$ into n-qubit registers, and that we can neglect the errors. In general we would need to bound the errors arising from the encoding, but a precision of $\mathcal{O}(n)$ bits ensures that the errors are inverse-exponential in n and negligible for the rest of the computations.

The key operation in Algorithm 3 is the comparison at Step 4. Without it, register 1 would remain disentangled from the others and the second H layer (Step 5) would simply map it back to $|0\rangle$.

After the comparison, when performing the Hadamard layer, any $|y\rangle$ such that $y/2^n < m(k^{\text{in}}, x)/G$ is mapped to $\frac{1}{2^{n/2}} |0\rangle + |*\rangle$ where $|*\rangle$ is a (non-normalized) superposition of non-zero basis states. Any other y is mapped to a superposition of non-zero basis states

Algorithm 3 Implementation of fFourier using a comparison, from [SLSB19].

1: Start in the state: $\underbrace{|k^{\text{in}}\rangle\,|x\rangle}_{\text{Register 0}}\ \underbrace{|0_n\rangle}_{\text{Register 1}}\ \underbrace{|0\rangle}_{\text{Register 2}}\ \underbrace{|0_n\rangle\,|0_n\rangle}_{\text{Ancillas}}$
2: Compute $\widehat{f}_{k^{\text{in}}}(x)/G$: $|k^{\text{in}}\rangle\,|x\rangle\,|0\rangle\,|0\rangle\,|m(k^{\text{in}},x)/G\rangle\,|\theta(k^{\text{in}},x)\rangle$
3: Apply H on register 1: $|k^{\text{in}}\rangle\,|x\rangle\,\frac{1}{2^{n/2}}(\sum_y|y\rangle)\,|0\rangle\,|m(k^{\text{in}},x)/G\rangle\,|\theta(k^{\text{in}},x)\rangle$
4: Compare $y/2^n$ and $m(k^{\text{in}},x)/G$: write 1 in register 2 if $y/2^n \geq m(k^{\text{in}},x)/G$
5: Apply H on register 1
6: Apply a rotation of angle $\theta(k^{\text{in}},x)$, controlled on registers 1 and 2
7: Uncompute $\widehat{f}_{k^{\text{in}}}(x)$

(since register 2 is 1). So in the second to last step, one obtains a superposition of the form:

$$|k^{\text{in}}\rangle\,|x\rangle\left(\frac{|\{y, y/2^n < m(k^{\text{in}},x)/G\}|}{2^n}\,|0\rangle\,|0\rangle + |*\rangle\right)|m(k^{\text{in}},x)/G\rangle\,|\theta(k^{\text{in}},x)\rangle \tag{26}$$

where $\frac{|\{y,y/2^n<m(k^{\text{in}},x)/G\}|}{2^n}$ is exactly the fixed-point approximation of $m(k^{\text{in}},x)/G$ that we use here. Finally the controlled rotation transforms this into:

$$|k^{\text{in}}\rangle\,|x\rangle\left(\exp(\iota\theta(k^{\text{in}},x))\frac{m(k^{\text{in}},x)}{G}\,|0\rangle\,|0\rangle + |*\rangle\right)|m(k^{\text{in}},x)/G\rangle\,|\theta(k^{\text{in}},x)\rangle \quad . \tag{27}$$

We can then erase the value $\widehat{f}_{k^{\text{in}}}(x)$ by calling the inverse of the digital computation unitary. The key operation here is the comparison at Step 4. □

In particular, if the input space of $\widehat{f}$ is small enough, the function can be precomputed and stored in a QRACM table. This makes the implementation of fFourier efficient, and the complexity of Corcomp will remain dominated by the queries to the database. This will be the case in our applications.

3.5 Simple Quantum Key-recovery

At this point, one has implemented the unitary that constructs the correlation state:

$$|k^{\text{in}}\rangle\,|0\rangle \xmapsto{\mathsf{Corcomp}} |k^{\text{in}}\rangle\,|\mathsf{Cor}_{k^{\text{in}}}\rangle \quad .$$

If the key-schedule is simple enough, one can append to this a uniform superposition over the remaining key bits, obtaining a superposition of possible master keys:

$$\sum_z \alpha_z\,|z\rangle$$

where the amplitude on the right key k can be larger, depending on the corresponding correlation. Then, one appends to this algorithm a test of z, which is the same as in Grover's search: one encrypts a few known plaintexts and sees if they match the expected results. This creates a unitary that produces a superposition $\sum_z \alpha_z\,|z\rangle\,|(z \stackrel{?}{=} k)\rangle$.

Amplitude Amplification (Theorem 1) can immediately be applied. Since one expects the starting amplitude for k to be larger than in an exhaustive search, less iterates will have to be performed. This advantage with respect to Grover's search leads to the attacks presented in [Sch23].

Obviously this approach has two main limitations:

- The speedup is limited by the experimental correlation. The smaller the correlation, the smaller the speedup with respect to exhaustive search. However, in classical linear cryptanalysis, the correlations tend to be the smallest possible (since one only wishes to distinguish the right key from the wrong ones), and this makes the speedup vanish.
- "Completing" the key only works if the key schedule is simple, and it is not clear if this technique can always be competitive with a quantum linear attack that does not use the QFT [KLLN16b].

In the next section, we show how to remove this second issue.

4 Improved Linear Attack using a Distinguisher

In this section, we depart from the framework of [Sch23]. We notice the following generalization: after computing the correlation state, what we really need is a *distinguisher* that will recognize the right key in the superposition. "Completing the key" as in [Sch23] is only a specific way to distinguish.

More formally, we prove the following result.

Theorem 3. *Let* $c = \sqrt{\mathrm{ELP}}$. *Let* Dist *be a unitary such that:*

$$|z^{\mathrm{in}}, z^{\mathrm{out}}\rangle\,|b\rangle \xmapsto{\mathsf{Dist}} \begin{cases} |z^{\mathrm{in}}, z^{\mathrm{out}}\rangle\,|b\oplus 1\rangle & \text{if } z^{\mathrm{in}}, z^{\mathrm{out}} = k^{\mathrm{in}}, k^{\mathrm{out}} \\ |z^{\mathrm{in}}, z^{\mathrm{out}}\rangle\,|b\rangle & \text{otherwise} \end{cases} \tag{28}$$

Then there is a quantum algorithm that uses on average less than

$$2\left\lceil 1.21\frac{2^{|k^{\mathrm{in}}|/2}}{c-2/\sqrt{N}}\frac{2^{n/2}}{\sqrt{N}}\right\rceil + 2 \tag{29}$$

calls to Dist *and* Corcomp *and returns* $k^{\mathrm{in}}, k^{\mathrm{out}}$ *with probability* ≥ 0.5.

Proof. The algorithm is a QAA over $(k^{\mathrm{in}}, k^{\mathrm{out}})$ that uses Corcomp to produce the superposition of candidate keys and Dist to identify the correct key.

The unitary Corcomp produces a superposition of key guesses:

$$\sum_{z^{\mathrm{in}}, z^{\mathrm{out}}} \alpha_{z^{\mathrm{in}}, z^{\mathrm{out}}}\,|z^{\mathrm{in}}, z^{\mathrm{out}}\rangle \quad . \tag{30}$$

By calling Dist immediately after, we obtain the superposition

$$\sum_{z^{\mathrm{in}}, z^{\mathrm{out}}} \alpha_{z^{\mathrm{in}}, z^{\mathrm{out}}}\,|z^{\mathrm{in}}, z^{\mathrm{out}}\rangle\,|b_{z^{\mathrm{in}}, z^{\mathrm{out}}}\rangle \quad . \tag{31}$$

By correctness of the distinguisher, $(k^{\mathrm{in}}, k^{\mathrm{out}})$ is the only element with a flag $b_{z^{\mathrm{in}}, z^{\mathrm{out}}} = 1$. Furthermore, by Equation 17, it has amplitude:

$$|\alpha_{k^{\mathrm{in}}, k^{\mathrm{out}}}| \geq (c - 2/\sqrt{N})2^{-|k^{\mathrm{in}}|/2}\frac{\sqrt{N}}{2^{n/2}} \quad . \tag{32}$$

We use QAA to boost the amplitude of the right key. As we only know a lower bound on the amplitude, we use Lemma 5. The unitary U from this lemma is simply the composition of CorComp and Dist, which produces the state from Equation 31. From Equation 32, $\sqrt{p_{\min}} = (c - 2/\sqrt{N})2^{-|k^{\mathrm{in}}|/2}\frac{\sqrt{N}}{2^{n/2}}$. Hence, Lemma 5 produces the wanted result. □

The above lemma shows that we obtain a key-recovery attack once we have an implementation of Dist. To implement it, we need a distinguisher on E_M in Figure 3, which highly depends on the structure of E_M. If a periodic function can be derived from E_M, we could apply an efficient Simon-based distinguisher.

Even if no Simon-based distinguisher is available, a linear distinguisher can always be applied. This is because, in the first place, E_M is assumed to have the linear approximation. We will elaborate the details in Subsection 4.3.

4.1 Attack Blueprint

Intuitive summary. We need two ingredients in our attacks:

- a high-correlation linear trail, that we use to create a quantum state *biased* towards the correct key,
- a distinguisher (that can be linear, a Grover-meet-Simon or anything else) to efficiently *identify* the correct key.

While using two kind of distinguishers on the same cipher with the same number of rounds may seem redundant, we can beat both pure quantum linear attacks and pure other distinguisher-based attacks when the conditions are right.

For the attack, we add rounds before and after the distinguisher part. Then, it is essentially a quantum amplitude amplification over the keys in the added rounds using the distinguisher, except that instead of starting from the uniform superposition of keys, we start from a state that has a higher amplitude for the correct key, thanks to the high-correlation trail.

Attack framework. Each attack presented in this paper relies on Theorem 3 and follows the same steps, which we summarize here.

First, we need to find the path of the attack:

1. We separate the cipher E into the key-recovery rounds and the rounds for the linear approximation (E'), such that there exists a high-correlation linear trail, and we determine a lower bound on the corresponding ELP;
2. We choose a distinguisher for E', not necessarily related to the linear approximation;
3. For the key-recovery rounds, we separate the subkey into k^{out} (which goes into the FFT) and k^{in} (which is guessed);
4. We express the experimental correlation $\widehat{\text{cor}}(z)$ as a convolution of two functions f and g.

Second, we perform the complexity analysis:

1. We estimate the complexity of Corcomp. Following Lemma 6, it depends mainly on:
   - The bound G on the Fourier coefficients of the function f. If possible, we give an exact bound. Most often, we estimate it by making heuristic assumptions.
   - The cost of the fFourier unitary. In the attacks of this paper, we implement fFourier via Lemma 7 and precomputed tables for the Fourier coefficients of f.
2. We estimate the complexity of Dist.
3. We use the formula of Theorem 3 to conclude.

4.2 Comparison with Other Techniques and Observations

This subsection compares the key-recovery attack of Theorem 3 with other techniques and obtains some important observations. We focus on the case where the full codebook of the target cipher is available in QRACM and $N(= |\mathcal{D}|) \simeq 2^n$ for simplicity.

We assume that a non-trivial linear approximation of E_M in Figure 3 (with $\sqrt{ELP} = c$ for some $c \gg 2^{-n/2}$) exists, and that an implementation of Dist of Equation 28 is available. The implementation may be based on a very efficient Simon-based distinguisher, a linear distinguisher we will explain in Subsection 4.3, or another algorithm. We do not specify it here.

The complexity of the key-recovery attack shown in Theorem 3 is about

$$\frac{2^{|k^{\text{in}}|/2}}{c}\big(\mathcal{G}(\mathsf{Dist}) + \mathcal{G}(\mathsf{Corcomp})\big).$$

Here, an important consideration is that the term $\mathcal{G}(\mathsf{Corcomp})$ could be as small as or even smaller than $\mathcal{G}(\mathsf{Dist})$ by performing some precomputations. In this case the complexity becomes about

$$\frac{2^{|k^{\text{in}}|/2}}{c} \cdot \mathcal{G}(\mathsf{Dist}). \tag{33}$$

Meanwhile, if a distinguisher on E_M is available, we can also mount a different key-recovery attack that just applies the Grover search on k^{in} and k^{out}, using the distinguisher to check if a given pair $(z^{\text{in}}, z^{\text{out}})$ matches $(k^{\text{in}}, k^{\text{out}})$. Since k^{out} is n-bit, the complexity of this attack is about

$$2^{(|k^{\text{in}}|+n)/2} \cdot \mathcal{G}(\mathsf{Dist}). \tag{34}$$

Since the factor $1/c$ is smaller than $2^{n/2}$ by assumption, we obtain the following observation.

Observation 1. *If $\mathcal{G}(\mathsf{Corcomp}) \ll \mathcal{G}(\mathsf{Dist})$, then the key-recovery attack combining FFT and QFT (Theorem 3 / Equation 33) is always faster than the attack searching for $(k^{\text{in}}, k^{\text{out}})$ just by using Grover's algorithm (Equation 34), regardless of what kind of distinguisher is available for E_M. Especially, if Dist is realized with a quantum linear distinguisher, the attack of Theorem 3 is faster than the quantum linear key-recovery attack that does not use QFT [KLLN16b], so the second issue mentioned at the end of Subsection 3.5 is resolved.*

This simple but important observation is not noted in [Sch23], and we emphasize that this paper is the first to point it out. The attack on CAST-256 in Section 8 is mounted with this observation in mind.

Furthermore, since the classical FFT key recovery requires the complexity about $2^{k^{\text{in}}} \times n2^n$ (see Subsection 3.2), we also have the following observation.

Observation 2. *Suppose $\mathcal{G}(\mathsf{Corcomp}) \ll \mathcal{G}(\mathsf{Dist})$ and $\mathcal{G}(\mathsf{Dist})/c \ll 2^{n/2}$. Then, with the key-recovery attack combining FFT and QFT (Theorem 3 / Equation 33), we achieve a super-quadratic speed-up compared to the classical FFT key-recovery attack.*

We have not found a concrete example demonstrating such a super-quadratic speed-up, but this observation could be a basis of very powerful quantum attacks in future works.

Remark 5. Since the right key has amplitude $1/c$ in the correlation state, another possibility is to measure directly and find the key that appears most often. This can be combined with a Grover's search on the inner key, leading to a complexity: $\widetilde{\mathcal{O}}\Big(\frac{2^{|k^{\text{in}}|/2}}{c^2}\mathcal{G}(\mathsf{Corcomp})\Big)$. However the computation of $\mathsf{Corcomp}$ is typically non negligible, and in practice this would lead to a much higher complexity.

4.3 How to Realize Dist with a Linear Distinguisher

This subsection discusses the details about how to implement Dist as a (quantum) linear distinguisher. The implementation is generic in that it can always be applied regardless of the structure of E_M in Figure 3 because E_M is assumed to have a linear approximation.

For simplicity, we focus on the case where the quantum oracle of a target cipher is available. That is, the attack model is Q2, or the model is Q1 but the full codebook of the cipher has been stored in QRACM and the quantum oracle is efficiently simulatable. We denote the cost to encrypt once with E_K (i.e., the gate count to implement the cipher on a quantum circuit) by Q.

Kaplan et al. [KLLN16b] have already shown how to speed-up a classical distinguisher using the quantum counting algorithm, so we could use it to realize Dist. However, they do not provide detailed analysis of the distinguisher's errors. Quantum algorithms cannot be used as a subroutine of another quantum algorithm if the errors are too large, and we would like to provide as precise analysis as possible. Therefore, we modify Kaplan et al.'s distinguisher so that the errors will be small enough.

Specifically, we run multiple instances of the quantum counting algorithm and then perform a majority vote, unlike Kaplan et al.'s running a single instance. The idea of running multiple instances and performing majority vote are quite similar to those of JDG in [Hos24].

Define a Boolean function $f_{z^{\mathrm{in}},z^{\mathrm{out}}}:\{0,1\}^n\to\{0,1\}$ by $f_{z^{\mathrm{in}},z^{\mathrm{out}}}(x)=1$ iff $\alpha\cdot F^L_{z^{\mathrm{in}}}(x\oplus z^{\mathrm{out}})=\beta\cdot F^R_{z^{\mathrm{in}}}(E_K(x))$. Let q, ℓ, and T be parameters (fixed later), and consider to run the following quantum algorithm without intermediate measurements, assuming $|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle$ is given as an input (b is a single bit to which the result of the distinguisher is added).

Algorithm 4 Algorithm LinDist.

1: **for** $i=1,\dots,\ell$ **do**
2: Run QC_q of Subsection 2.6 to estimate $|f^{-1}_{z^{\mathrm{in}},z^{\mathrm{out}}}(1)|$. Denote the result by $\tilde{X}_i$.
3: **end for**
4: If the number of indices i satisfying $|2\tilde{X}_i/2^n-1|\geq T$ is greater than or eqal to $\ell/2$, XOR 1 to b (meaning that the algorithm judges $(z^{\mathrm{in}},z^{\mathrm{out}})=(k^{\mathrm{in}},k^{\mathrm{out}})$). Otherwise, do nothing (meaning that the algorithm judges $(z^{\mathrm{in}},z^{\mathrm{out}})\neq(k^{\mathrm{in}},k^{\mathrm{out}})$).
5: Uncompute Step 1.

Then, the following lemma holds.

Lemma 8. *Let $\kappa:=|k_{in}|+|k_{out}|$. The gate count of the above algorithm is approximated by $8q\ell Q$. In addition, if $T=\frac{3}{8}\sqrt{ELP}$, $q=\left\lceil 16\pi/\sqrt{ELP}\right\rceil$, and $1\gg\sqrt{ELP}\geq 2^{-n/2}\cdot(4\sqrt{2\kappa})$ holds, then the following statement holds with probability of at least around $0.95-(2/e)^{\kappa}/\sqrt{2\kappa}$ (over the choice of the key): As a unitary operator, the algorithm* LinDist *approximates the unitary operator* Dist $:|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle\mapsto|z^{\mathrm{in}},z^{\mathrm{out}},b\oplus((z^{\mathrm{in}},z^{\mathrm{out}})\stackrel{?}{=}(k^{\mathrm{in}},k^{\mathrm{out}}))\rangle$ *with an error in $2^{(|k^{\mathrm{in}}|+|k^{\mathrm{out}}|)/2-0.045\ell+2}$ with respect to the operator norm.*

Suppose a quantum algorithm calls Dist as a subroutine r times, and LinDist is used as a distinguisher to approximate Dist. In such a case, by setting

$$\ell:=\frac{1}{0.045}\left(\frac{|k^{\mathrm{in}}|+|k^{\mathrm{out}}|}{2}+\log_2 r+2n\right), \tag{35}$$

this lemma guarantees that the errors caused by replacing Dist with LinDist are kept within $O(2^{-2n})$ and can be ignored (with a high probability over the choice of the key). The total gate count of LinDist with this ℓ is in $\mathcal{O}\Big((|k^{\mathrm{in}}|+|k^{\mathrm{out}}|+\log_2 r+n)Q/\sqrt{ELP}\Big)$.

A proof of the lemma can be found in Appendix A.

Remark 6. A previous work [Hos24] also utilizes correlations for both preparing a quantum state and amplifying a correct value, but it is in the context of fast correlation attacks on LFSR-based stream ciphers.

5 Quantum Linear Cryptanalysis of LOKI91

In this section we use LOKI91 as an example for applying both the framework of [Sch23] and our new extensions. Our results and the comparison with the state of the art are summarized in Table 1.

We start with the description of a new attack on a round-reduced version of LOKI91 using the framework of [Sch23]. Next, we introduce an alternative approach applying the strategy summarized in Subsection 4.1 based on a Simon-based distinguisher. Finally, we discuss a third strategy, where we apply the linear distinguisher (LinDist).

5.1 Specification of LOKI91 and Summary of Results

LOKI91 [BKPS91] is a 64-bit block cipher with 64-bit key. It is a 16-round Feistel network. The 64-bit master key k is divided into 32-bit halves k^l and k^r, which are then processed to generate the sequence of 16 round keys:

- After every round r, the corresponding round key k_r is rotated whereby the rotations alternate between 13 bits (ROL 13) and 12 bits (ROL 12).
- After every second round, the halves are swapped.

In particular, we observe that $k_1 = k^l$ and $k_3 = k^r$. For each round r, let y_r be the right half of the input. The round key k_r is XORed with y_r and the result is passed as input to the non-linear round function F defined as

$$F = P(S(E(y_r \oplus k_r))),$$

where E expands the 32-bit input to 48 bits, S is an S-Box layer of four identical 12-to-8 bit S-Boxes and P is a permutation. We omit further details that are not relevant for our analysis.

We attack a 6-round version of LOKI91 as represented in Figure 4. Obviously the cipher itself is insecure due to a too small key length; besides, many more rounds can be broken by linear cryptanalysis [SF97]. Our goal is only to compare different quantum linear attacks on this particular example.

Table 1: Classical and quantum attacks on 6-round LOKI91. "Q0" indicates a classical attack. Quantum time complexities are counted in equivalent computations of the cipher.

Setting	Type	Rounds	Data	Time	QRACM	Reference
Q0	Linear	6	$2^{32.2}$	2^{51}		[TSM94]
	Linear	6	2^{23}	2^{37}		[ÖBR23]
	Linear	10	$2^{54.83}$	$2^{54.83}$		[SF97]
Q1	Grover's search	6	2	$2^{33.65}$	None	
Q1	Linear + QFT + Grover	6	2^{32}	$2^{32.78}$+ 2^{32} classical	2^{32}	Sec. 5.2
Q2	Linear + QFT + Simon	6	$2^{23.71}$	$2^{23.73}$+ 2^{32} classical	2^{32}	Sec. 5.3

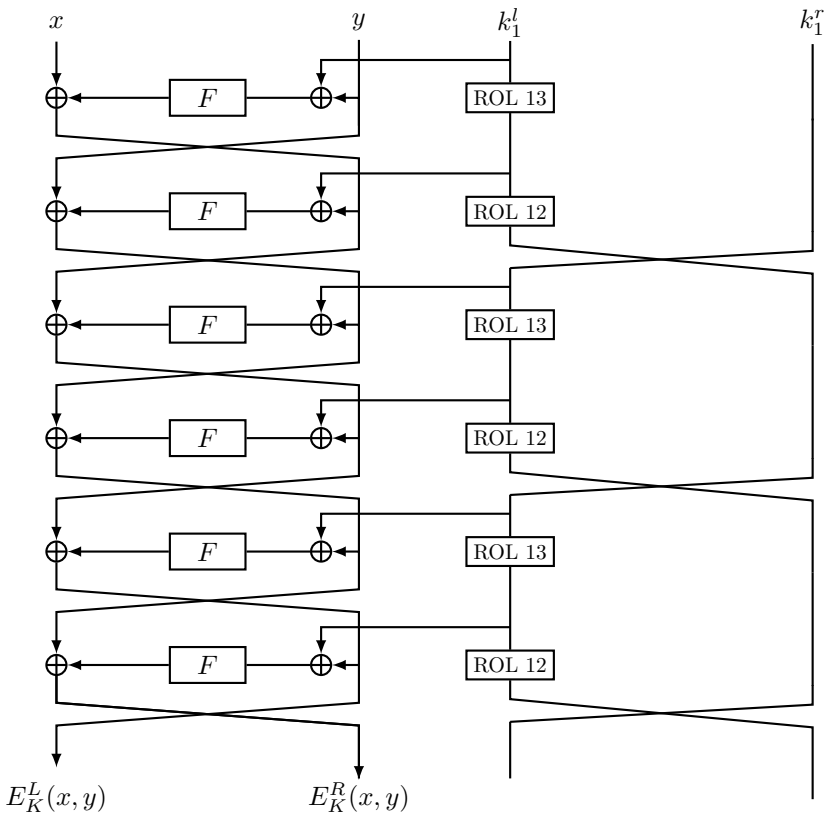

Figure 4: 6 rounds of LOKI91

Linear approximation. All attacks presented in this section rely on the linear approximation on 5-round LOKI91 from [TSM94], which holds with probability $\frac{1}{2} - 1.6 \times 2^{-15}$. The input mask has only three non-zero bits (18,22,26) in the right branch. The output mask has three non-zero bits (18,22,26) in the left branch, and 5 nonzero bits (18,19,21,23,24) in the right branch.

By the consequence of the right-key hypothesis (Equation 17), we can assume that the corresponding correlation is bounded from below by $2 \times 1.6 \times 2^{-15} - 2 \times 2^{-16} = 2^{-13.32} - 2^{-15} = 2^{-13.86}$.

In the following we apply this linear approximation for input plaintexts, whose left half is fixed to $0 \in \{0,1\}^{32}$. This is feasible due to the specific form of the linear approximation mask.

5.2 Attacking LOKI91 using the strategy from [Sch23]

In the following, the complexity is counted in queries to the cipher. A large QRACM access is assumed to cost the same, and this quantity is denoted Q.

Path of the Attack. Our first attack on LOKI91 follows the strategy from [Sch23] outlined in Subsection 3.5. Due to the simple key schedule of LOKI91, this strategy is successful. That is, we obtain an attack, which is more efficient than a Grover search for the 64-bit masterkey. Note that a Grover search to find the 64-bit-masterkey has the cost $\frac{\pi}{4} \times 2^{32} \times 4Q \simeq 2^{33.65}Q$ (4 block cipher calls per iteration).

Experimental Correlation. Let $n = 64/2 = 32$. Define:

$$\widehat{\mathrm{cor}}(z) = \frac{1}{2^n} \sum_{y \in \{0,1\}^n} (-1)^{\alpha \cdot F(z \oplus y)} (-1)^{\beta \cdot E_K(0,y)}$$

where α, β correspond to the linear mask associated to the linear approximation mentioned above, and we denote by $E_K(0, y)$ the output of the 6-round LOKI91 block cipher (the left half of the plain text being given by $0 \in \{0,1\}^n$ and the right half by $y \in \{0,1\}^n$).

Note that here, in contrast with some other parts of this paper (especially Figure 1 and Figure 3), n is equal to half of the block size of the cipher. This is due to the fact that here, for the experimental correlation, we fix half of the input to the cipher. However, on the other hand, this way we are aligned with the notation in Equation 14, and thus also with the notation used in Lemma 6 and Theorem 3, which we apply in this section.

We introduce the two functions f and g as

$$\begin{cases} f(y) = (-1)^{\alpha \cdot F(y)} \\ g(y) = (-1)^{\beta \cdot E_K(0,y)} \end{cases}$$

Then, the experimental correlation is: $\widehat{\mathrm{cor}}(z) = \frac{1}{2^n}(f \star g)(z)$.

Computation of the Correlation State. We follow Lemma 6 to construct the correlation state via the unitary Corcomp. For a technical reason which is detailed later, we swap the roles of the functions f and g. This means that we need to precompute $\widehat{g}$, using 2^{32} classical queries and somewhat more classical computations. We store $\widehat{g}$ in a QRACM of size 2^{32}.

Recall that, according to Lemma 6, the unitary performs $\left(\frac{\pi}{2} G/2^{n/2} + 3\right)$ iterations where $G = \max_x |\widehat{g}(x)|$. Each iteration contains one computation of f and an amplitude transduction with $\widehat{g}$, which contains two QRACM queries.

Assuming that g behaves as a random function, we can follow [Sch23, Lemma 10] (presented in Appendix B) showing that G is $\mathcal{O}\left(\sqrt{n} 2^{\frac{n}{2}}\right)$. The cost of each iteration is dominated by the QRACM queries. Overall the complexity is such that CorComp does not dominate in the rest of our analysis, and can be neglected.

Cost of the Distinguisher. It remains to implement the distinguisher Dist for k^l. As already mentioned, in this subsection, we follow the strategy of [Sch23] outlined in Subsection 3.5. That is, we perform a Grover's exhaustive search of k_1^r, the right half of the 64-bit masterkey. Indeed, once k_1^r is known we have the full masterkey, and we can test if it's correct using two plaintext-ciphertext pairs. For a given key z_1^l, checking if there exists z_1^r such that (z_1^l, z_1^r) is the correct master key requires to perform $\frac{\pi}{4} 2^{16}$ search iterates, with 2×2 block cipher computations per iterate.

Cost of the Attack. According to Theorem 3 we know that there is a quantum algorithm that makes less than

$$2 \left\lceil 1.21 \frac{1}{2^{-13.86}} \right\rceil + 2 \simeq 2^{15.13} \tag{36}$$

calls to Corcomp and the just-described distinguisher Dist, where we used that $|\widehat{\mathrm{cor}}(k^l)| \geq 2^{-13.86}$. Hence, we obtain a total time equivalent to:

$$2^{15.13} \times \frac{\pi}{4} 2^{16} \times 2^2 Q \simeq 2^{32.78} Q \ . \tag{37}$$

This is slightly below the number of computations required for exhaustive Grover key search.

5.3 Improved Linear Attack using a Simon-based Distinguisher

In this subsection we follow the approach outlined in Subsection 4.1 by using a Simon-based distinguisher. Note that the experimental correlation and the cost of preparing the correlation state are the same as in Subsection 5.2.

Cost of the Distinguisher. It starts from a 4-round generic distinguisher on a Feistel network using Simon's algorithm, which is described in [IHM+19]; cf. Subsection 2.4. This distinguisher requires access to both the cipher and its inverse. By using $96 = 2^{6.58}$ queries in Simon's algorithm (hence $2^{7.58}$ Q2 queries to the cipher), we ensure a large success probability.

From this distinguisher, we implement a unitary SimonDist that given a guess for the key k^l, unrolls the first two rounds (in which only k^l intervenes) and distinguishes the remaining 4 rounds. The cost of this unitary:

$$|z\rangle\,|b\rangle \xmapsto{\mathsf{SimonDist}} |z\rangle\,|b \oplus (z \stackrel{?}{=} k^l)\rangle$$

can be approximated by $2^{8.58}Q$.

Cost of the Attack. The cost of the attack up to now can be estimated to:

$$2^{15.13} \times 2^{8.58}Q \simeq 2^{23.71}Q \ . \tag{38}$$

For this attack it does not seem possible to get rid of Q2 queries, as we need to craft specific inputs to the 4-round reduced cipher for the Simon-based distinguisher. However, we still need to make 2^{32} classical queries, and precomputations, in order to construct easily the correlation state.

Note that, up to now, this attacks recovers only k^l. An additional Grover search yields also k^r. The total cost of key-recovery becomes:

$$2^{23.71}Q + \frac{\pi}{4}2^{16} \times 2^2 Q = 2^{23.71}Q + 2^{17.65}Q = 2^{23.73}Q \ . \tag{39}$$

The difference is modest since we have already recovered a good proportion of the key. In the other attacks of this paper, we will omit this last step.

5.4 Remark about using LinDist as the Distinguisher

As a last attempt, we use a distinguisher based on the linear distinguisher LinDist from Subsection 4.3. It is interesting to see that LinDist is *not* the right distinguisher to use for LOKI91. The first reason for that is that the condition $1 \gg \sqrt{ELP} \geq 2^{-n/2} \cdot (4\sqrt{2\kappa})$ in Lemma 8 is not fulfilled for the given linear approximation. The second reason is that, although we know from Section 4 that this attack strategy will be more efficient asymptotically (as the complexity will go down from $\mathcal{O}(2^{n/2})$ to $\mathcal{O}(c^{-2})$), it turns out that the constant factors that are contained in the complexity estimation of Lemma 8 are too large (as we can see in the following).

Cost of the Distinguisher. According to Lemma 8 the cost of this distinguisher is given by $8q\ell Q$, where

$$q = \lceil 16\pi/c \rceil \simeq 2^{18.97}, \quad \ell := \frac{1}{0.045}\left(\frac{|k^{\text{out}}|}{2} + \log_2 r + n\right) = \frac{1}{0.045}\left(\frac{5}{4}n + \log_2 r\right) \simeq 2^{11.04}$$

where

$$r = 2\left\lceil 1.21\frac{1}{c}\right\rceil + 2 \simeq 2^{14.59} \ .$$

That is, the complexity of LinDist becomes

$$8q\ell Q = 2^3 \times 2^{18.97} \times 2^{11.04} Q = 2^{32.01} Q \ ,$$

which is far too expensive.

5.5 Remark on the Fourier Transforms

The swapping of f and g limits greatly this particular application, since an incompressible cost of 2^{32} classical queries, classical computations, and QRACM, needs to be taken into account.

The reason for this tweak is the following. If we try to follow directly our blueprint, we need to perform amplitude transduction of the function: $f(y) = (-1)^{\alpha \cdot F(y)}$. To apply Lemma 6, we need a bound G on $|\widehat{f}(y)|$. Contrary to g, f cannot be modeled as a random function. To analyze it precisely, we must look into the round function F.

In LOKI91, F contains an expansion E (which permutes and copies some of the bits) from 32 to 48 bits, followed by a layer of 4 S-Boxes mapping 12 bits to 8 bits each, followed by a bit permutation. It can be noticed that for the choice of α that we took from [TSM94], $\alpha \cdot F(y)$ depends only on a single S-Box, hence only on 12 bits of the input y.

This is actually a problem for the attack. Indeed, the function f is far from a random Boolean function on 32 bits: it has many zero Fourier coefficients, and the nonzero ones are very large. Therefore the bound G becomes much larger as well, making the Corcomp procedure inefficient.

6 New Quantum Attack on CAST-128

In this section, we give a quantum attack on CAST-128 reduced to 7 rounds using linear cryptanalysis and FFT. Our results are summarized in Table 2.

As explained in Subsection 2.2, we let Q be the cost (gate count) of a 7-round CAST-128 computation, and both a Q2 query and a QRACM query are assumed to cost Q. As a consequence the cost of Grover's search can be lower bounded by: $\frac{\pi}{4} 2^{128/2} \times 2 \times 2Q = 2^{65.7} Q$ (each iteration requires two block cipher computations and their uncomputation).

The cost of a partial computation of the cipher, as well as a QRACM query to a smaller database, will be upper bounded by Q. At some point, we will also need to compare Q to an actual gate count. Neglecting the key schedule, we estimate that the cipher contains $(4 + 7 \times 2 + 7 \times 4) = 2^{5.5}$ 32-bit operations (modular addition, subtraction) and that each of them contains at least $64 = 2^6$ quantum gates. Therefore we have $Q \geq 2^{11.5}$.

6.1 Specification of CAST-128 and Summary of Results

We start here by summarizing the design of the CAST-128 cipher. We focus only on the parts that are relevant for our analysis, and ignore the key-schedule (we consider all round keys to be independent).

CAST-128 [Ada97a, Ada97b] is a 64-bit block cipher with up to 128 bits of key (we consider only the 128-bit key version). It is a 16-round Feistel network, as represented in Figure 5. The round function alternates between three designs F_1, F_2, F_3, which are not permutations. Each round uses 37 bits of subkey material: a 5-bit value k_i^r which controls a shift, and a 32-bit one k_i^m which is alternatively added, XORed or subtracted in the order that can be seen on the figure.

The definition of F_1, F_2 and F_3 is as follows. Let X be the branch value before key addition and $I = I_a|I_b|I_c|I_d$ be the value after key addition with least significant byte to most significant byte. CAST uses 4 S-Boxes S_1, S_2, S_3, S_4 mapping a byte to 32 bits.

$$\begin{array}{lll} \text{Type 1} & I = ((k_i^m + X) \lll k_i^r) & F_1(I) = ((S_1[I_a] \oplus S_2[I_b]) - S_3[I_c]) + S_4[I_d] \\ \text{Type 2} & I = ((k_i^m \oplus X) \lll k_i^r) & F_2(I) = ((S_1[I_a] - S_2[I_b]) + S_3[I_c]) \oplus S_4[I_d] \\ \text{Type 3} & I = ((k_i^m - X) \lll k_i^r) & F_3(I) = ((S_1[I_a] + S_2[I_b]) \oplus S_3[I_c]) - S_4[I_d] \end{array}$$

The best classical attack targets 8 rounds, using a 5-round multidimensional linear distinguisher [ÖBR23] and 3 rounds for key-recovery. A meet-in-the-middle attack given in [IS13] also allows to target 8 rounds.

In the quantum setting, it is possible to attack 7 rounds of CAST-128 using the Grover-meet-Simon algorithm [LM17]. One will guess 3 subkeys, for a total of 111 bits. The remaining 4-round Feistel can be distinguished from a random permutation in polynomial time with Simon's algorithm of [IHM+19]. We estimate (Subsection 6.6) that this attack has complexity $2^{64.5}Q$, with Q2 queries or QRACM.

Table 2: Classical and quantum attacks on CAST-128. "Q0" indicates a classical attack. Quantum time complexities are counted in equivalent computations of the corresponding cipher.

Setting	Type	Rounds	Data	Time	QRACM	Reference
Q0	MITM ASR	7	6	2^{114}		[IS12]
	Linear	6	$2^{53.96}$	$2^{88.51}$		[WWH08]
	MITM ASR	8	8	2^{118}		[IS13]
	Linear	8	2^{35}	2^{114}		[ÖBR23]
Q1	Grover's search	7	2	$2^{65.7}$	None	
Q2	Grover-meet-Simon	7	$2^{64.5}$	$2^{64.5}$	None	
Q1	Grover-meet-Simon	7	2^{64}	$2^{64.5}$	2^{64}	
Q1	Linear + QFT + Simon	7	2^{64}	$2^{61.1}$	2^{64}	Section 6

In this section, we use the combination of a quantum linear attack using the QFT, and a distinguisher based on Simon's algorithm, to reduce the complexity of the 7-round attack on CAST-128.

6.2 Path of the Attack and Distinguishers

The path of the attack is represented in Figure 5. It combines two rounds of FFT-based key-recovery (first and last round in the figure), a 5-round linear distinguisher (5 middle rounds in the figure) and the 4-round Simon-based distinguisher of [IHM+19] (see Subsection 2.4), which is extended to a 5-rounds distinguisher by guessing the 37 bits of key (k_2^r, k_2^m) (5 middle rounds in the figure).

For the linear distinguisher, we use a linear approximation holding for F_1 and F_3 at the same time. Following Table 1 in [WWH08], we find that a mask α activating only the 18-th bit allows to approximate both F_1 and F_3 with quite a large bias: $0 \to \alpha$ holds with bias $2^{-14.41}$ for F_1 (correlation $2^{-13.41}$) and bias $2^{-14.47}$ for F_3 (correlation $2^{-13.47}$).

By combining these two approximations, the approximation $(0, \alpha) \to (0, \alpha)$ holds for the 5 middle rounds in Figure 5 with correlation $2^{-26.88}$. For the right key guess, when using at least 2^{63} data, by Equation 17 we can lower bound the experimental correlation by $2^{-26.88} - 2 \times 2^{-63/2} \simeq 2^{-27.0}$.

6.3 Data

In the following, we denote $(x, y, E_K^L(x, y), E_K^R(x, y))$ the respective left and right hand sides of a plaintext and its corresponding ciphertext.

We start from a database $\mathcal{D}$ of known-plaintext queries, which is stored in QRACM. At first, we may consider $\mathcal{D}$ as a list of tuples $(x, y, E_K^L(x, y), E_K^R(x, y))$. However, in order

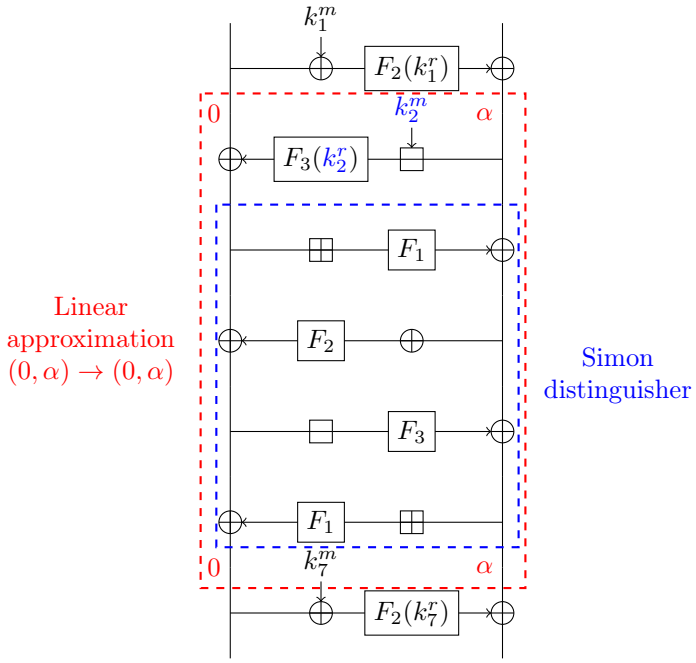

Figure 5: 7-round attack. The keys in blue are guessed to extend the Simon distinguisher.

to facilitate quick (and quantum) access to its elements, we may index the 4-tuples on any pair of values.

For example, if we index them on $(x, E_K^L(x,y))$, we can call a function:

$$(z,t) \mapsto \begin{cases} (y, E_K^R(x,y)) \text{ if } \exists (x,y) \in \mathcal{D}, z = x, t = E_K^L(x,y) \\ \perp \text{ otherwise} \end{cases}$$

In order to define this function, we are required to keep only one (arbitrary) corresponding entry.

If we start from the full codebook, we expect the function to be defined for approximately $(1-e^{-1})2^n \simeq 2^{n-0.66}$ values, which is the average number of image points in a random function from n to n bits.

Remark 7. We do not detail here how this data structure can be implemented in QRACM, and assume that the operations of creating superpositions over the data, and getting the data in superposition, are given.

By abuse of notation, we use $\mathcal{D}$ to denote any such function, e.g.:

$$\mathcal{D}(x, E_K^L(x,y)) = y, E_K^R(x,y) \ ,$$

which given an index specifying at most $\mathcal{O}(1)$ tuples in the database, arbitrarily selects one such tuple if it exists and returns the missing values. And in the following, we denote by $N := |\mathcal{D}| = 2^{n-0.66}$ the amount of available data (which is also the amount of data that we need to store in QRACM).

Remark 8. The construction of this function $\mathcal{D}$ is similar to the *distillation phase* used in classical cryptanalysis [FN20], except that we suffer here from additional requirements from the quantum setting: when there are collisions between tuples, we must drop some of them, and we cannot use all the data.

Experimental Correlation. In our attack, the *inner keys* are $k^{\text{in}} = k_1^r, k_7^r$ and the *outer keys* are $k^{\text{out}} = k_1^m, k_7^m$. We denote by $z_1^r, z_7^r, z_1^m, z_7^m$ a guess of these keys. We introduce the database $\mathcal{D}$ and use the function: $\mathcal{D}(x, E_K^L(x,y)) = (y, E_K^R(x,y))$ (over valid entries). We rewrite this as a function that, to any entry $(z,t) := (x, E_K^L(x,y))$, associates $(\mathcal{D}(z,t)^L, \mathcal{D}(z,t)^R) := (y, E_K^R(x,y))$. Summing over the database $\mathcal{D}$ (of size $N := |\mathcal{D}| = 2^{n-0.66}$), the correlation is:

$$\widehat{\text{cor}}(z_1^m, z_7^m, z_1^r, z_7^r) = \frac{1}{N}\sum_{z,t}(-1)^{\alpha\cdot(\mathcal{D}(z,t)^L \oplus F_2(z_1^r, z_1^m \oplus z))}(-1)^{\alpha\cdot(\mathcal{D}(z,t)^R \oplus F_2(z_7^r, z_7^m \oplus t))} \quad . \tag{40}$$

We define two functions $f_{z_1^r, z_7^r}$ and g as follows:

$$\begin{cases} g(z,t) = \mathbf{1}[(z,t) \in \mathcal{D}](-1)^{\alpha\cdot(\mathcal{D}(z,t)^R \oplus \mathcal{D}(z,t)^L)} \\ f_{z_1^r, z_7^r}(z,t) = (-1)^{\alpha\cdot(F_2(z_1^r, z) \oplus F_2(z_7^r, t))} \end{cases} \tag{41}$$

and the experimental correlation becomes:

$$\widehat{\text{cor}}(z_1^m, z_7^m, z_1^r, z_7^r) = \frac{2^n}{N}\left(f_{z_1^r, z_7^r} \star g\right)(z_1^m, z_7^m) \quad . \tag{42}$$

In the remainder of this section, we explain how we compute the correlation state (in superposition over z_1^r, z_2^r):

$$|\mathsf{Cor}_{z_1^r, z_7^r}\rangle := \frac{\sqrt{N}}{2^{n/2}}\sum_{z_1^m, z_7^m} \widehat{\text{cor}}(z_1^m, z_7^m, z_1^r, z_7^r)\,|z_1^m, z_7^m\rangle \quad . \tag{43}$$

Afterwards, we explain how this fits into the attack and finish the complexity analysis.

6.4 Computation of the Correlations

In order to compute $\mathsf{Cor}_{z_1^r, z_7^r}$ we need to implement the unitary:

$$|z_1^r, z_7^r\rangle\,|x,y\rangle\,|0\rangle \mapsto |z_1^r, z_7^r\rangle\,|x,y\rangle\left(\frac{\widehat{f}_{z_1^r, z_7^r}(x,y)}{G}\,|0\rangle + |*\rangle\right) \tag{44}$$

where G is an upper bound which remains to be determined, and $\widehat{f}_{z_1^r, z_7^r}(x,y)$ has the expression:

$$\widehat{f}_{z_1^r, z_7^r}(x,y) \quad = \quad \left(\sum_u (-1)^{x\cdot u}(-1)^{\alpha\cdot F_2(z_1^r, u)}\right) \quad \times \quad \left(\sum_u (-1)^{y\cdot u}(-1)^{\alpha\cdot F_2(z_7^r, u)}\right) \quad .$$

We will be able to determine G exactly, thanks to the simple definition of f. Indeed, let us focus on:

$$\begin{aligned} f'_{z_1^r}(x) &:= \sum_u (-1)^{x\cdot u}(-1)^{\alpha\cdot F_2(z_1^r, u)} \\ &= \sum_{\substack{w \in \mathbb{F}_2^{32} \\ (w = u \lll z_1^r)}} (-1)^{\alpha\cdot F_2(0,w)}(-1)^{(w \ggg z_1^r)\cdot x} \\ &= \sum_{u \in \mathbb{F}_2^{32}} (-1)^{\alpha\cdot F_2(0,u)}(-1)^{u\cdot(x \lll z_1^r)} \quad . \end{aligned}$$

We separate u into 4 bytes u_a, u_b, u_c, u_d and decompose F_2 with the 4 S-Boxes S_1, S_2, S_3, S_4 (which are functions from $\mathbb{F}_2^8$ to $\mathbb{F}_2^{32}$).

$$f'_{z_1^r}(x) = \sum_{u \in \mathbb{F}_2^{32}} (-1)^{\alpha \cdot S_4(u_d)} (-1)^{\alpha \cdot [(S_1(u_a) - S_2(u_b)) + S_3(u_c)]} (-1)^{u \cdot (x \lll z_1^r)} \quad . \tag{45}$$

In the end we see that $f'_{z_1^r}(x)$ is the product of a Walsh coefficient of an 8-bit function: $u_d \mapsto (-1)^{\alpha \cdot S_4(u_d)}$, and a Walsh coefficient of a 24-bit function: $u_a, u_b, u_c \mapsto (-1)^{\alpha \cdot [(S_1(u_a) - S_2(u_b)) + S_3(u_c)]}$. The value of z_1^r only changes which coefficient we select.

The function $u_d \mapsto (-1)^{\alpha \cdot S_4(u_d)}$ is bent (by property of the S-Boxes used in CAST). Therefore its Walsh coefficients have all the same absolute value. For the second function, we compute its Walsh coefficients experimentally and take the following upper bound:

$$\forall x, \forall z_1^r, |f'_{z_1^r}(x)| \leq 2^4 \times 2^{16} = 2^{20} \quad . \tag{46}$$

From which the upper bound: $G := 2^{40}$ follows.

Computation. Our goal now is to actually implement the operation of Equation 44. This is actually quite simple, since the output of $\widehat{f}_{z_1^r, z_7^r}(x, y)$ is the product of four Walsh coefficients of functions of at most 24 bits. Since we are using QRACM, we can precompute these functions and access the QRACM for their values.

Then we compute the four values in the amplitude independently using the method of Lemma 7. The amplitude over the full-zero state is indeed the product of these four values, and gives the wanted result. This required 8 QRACM lookups. The gate count is thus approximately $8Q$.

Using Lemma 6 with $\frac{\pi}{2} \frac{G}{2^{n/2}} \simeq 2^{8.7}$, we conclude that we can implement Corcomp with gate count: $2^{8.7}Q + 2^{11.7}Q \simeq 2^{11.9}Q$.

6.5 Cost of the Distinguisher

Assuming $k_1^m, k_1^r, k_7^m, k_7^r$ are known, the remaining 5 rounds in Figure 5 can be distinguished using an instance of the Grover-meet-Simon algorithm. The key guess is k_2^m, k_2^r (37 bits). Since the function is on 32 bits, by using $2^7 = 128$ queries in Simon's algorithm, we ensure to detect a periodic function with overwhelming probability.

Lemma 9. *There exists an implementation of* Dist :

$$|z_1^m, z_1^r, z_7^m, z_7^r\rangle |b\rangle \mapsto |z_1^m, z_1^r, z_7^m, z_7^r\rangle \begin{cases} |b \oplus 1\rangle & \textit{if guess is good} \\ |b\rangle & \textit{otherwise} \end{cases}$$

using gate count $\leq 2^{27.5}Q$.

Proof. Starting from $|z_1^m, z_1^r, z_7^m, z_7^r\rangle$, we perform a Grover search in a new register $|z_2^m, z_2^r\rangle$. The test in this search looks whether the cipher defined by removing the first two rounds, and the last one, is a 4-round Feistel or not. This is done using the distinguisher of [IHM+19] recalled in Subsection 2.4, where the periodic function contains one call to the cipher and one to its inverse.

After $\frac{\pi}{4} 2^{18.5}$ search iterates, we know that $|z_3^m, z_3^r\rangle$:

- Contains the key $|k_3^m, k_3^r\rangle$ with high probability if our guess was correct;
- Does not contain the key otherwise.

Therefore, we apply the Simon-based test again and flip the flag b depending on its result.

Each search iterate contains:

- $128 \times 2 = 2^8$ (QRACM) calls to E_K (for computing and uncomputing) and 2^8 calls to E_K^{-1}: cost $2^9 Q$
- $3 \times 2^9 \simeq 2^{10.6}$ CAST S-Boxes and 32-bit operations (approximately $2^5 Q$)
- 1 call to a quantum circuit that computes the rank of a 32×128 matrix, and flips b if this rank is maximal. Using the circuit of [BJ22], we need about $2 \times 32 \times 128 \times 32 = 2^{18}$ Toffoli gates. By our assumption on the value of Q, this is smaller than $2^{6.5} Q$.

The total cost can then be upper bounded by:

$$\frac{\pi}{4} 2^{18.5} \left(2^9 Q + 2^5 Q + 2^{6.5} Q\right) \simeq 2^{27.5} Q \ . \tag{47}$$

□

6.6 Cost of the Attack

We use Theorem 3 with:

$$k^{\text{in}} = k_1^r, k_7^r, \quad k^{\text{out}} = k_1^m, k_7^m, \quad |k^{\text{in}}| = 10, \quad |k^{\text{out}}| = 64 \ . \tag{48}$$

The entire key-recovery attack performs:

$$2 \left\lceil 1.21 \times 2^{27.0} \times 2^5 \times 2^{0.33} \right\rceil \simeq 2^{33.6} \tag{49}$$

calls to both CorComp and Dist to succeed with probability ≥ 0.5.

This gives the following gate count:

$$2^{33.6} \times (2^{11.9} Q + 2^{27.5} Q) \simeq 2^{61.1} Q \ ,$$

where we see that the cost of the Grover-meet-Simon distinguisher largely dominates.

As a comparison, the Grover-meet-Simon attack simply guesses 3 round keys and uses the same Simon-based distinguisher. Its complexity is approximately: $2^{37+27.5} Q = 2^{64.5} Q$.

Since the computation of Corcomp does not dominate in our attack, the gain comes entirely from having to perform less QAA iterates.

7 New Quantum Attack on CAST-256

We present here an attack on CAST-256 similar to the one of Section 6, which combines a Simon-based distinguisher with a linear approximation, and applies FFT-based key-recovery on part of the input and the output of the cipher.

In addition to Section 6, we use a generalized FFT. Indeed, in a quad-round starting from x, y, z, t, we can express the value of y after the first two rounds as:

$$y' = y \oplus F_2(k_2^r, k_2^m \oplus z \oplus F_1(k_1^r, k_1^m + t)) \ .$$

Thus we will use FFT for a function of $\mathbb{F}_2^n \times \mathbb{Z}_{2^n}$.

7.1 Specification of CAST-256 and Summary of Results

The CAST-256 [Ada97a, AG99] cipher is a 48-round type-1 Generalized Feistel scheme which was a candidate to the AES competition. It has a block size of 128 bits and a key length up to 256 bits (we will focus on the 256-bit version).

CAST-256 borrows the definition of the S-Boxes and the round functions F_1, F_2, F_3 from CAST-128 (see Subsection 6.1). We omit the description of its key schedule, and consider all round keys to be independent.

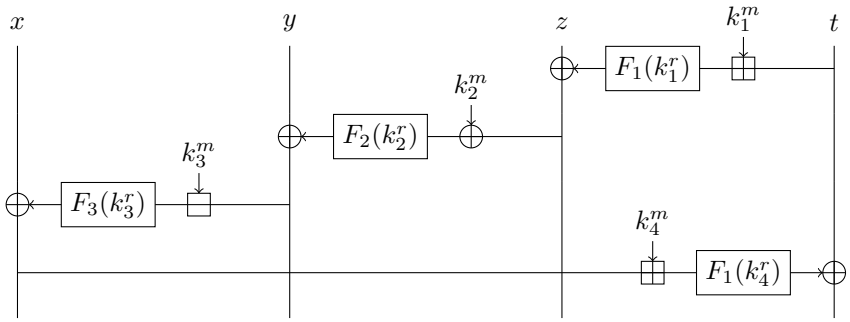

Figure 6: CAST-256 quad-round.

The rounds in CAST-256 are bundled in *quad-rounds* (Figure 6), so that the full cipher contains 6 quad-rounds, followed by 6 *reverse* quad-rounds, in which the order of the operations is reversed.

In Table 3, we recall a few results of classical and quantum cryptanalysis on CAST-256. The best linear attack reaches 32 rounds. In the quantum setting, the best known attack uses the Grover-meet-Simon algorithm [SCQ+23], where a distinguisher on 17-round CAST-256 is combined with a Grover's search for 6 subkeys. The complexity given in [SCQ+23] is 2^{111}. However, since we take into account the cost of Simon's algorithm, we increase it to $2^{123.7}Q$ (we use about 2^8 queries in each Simon subroutine).

Table 3: Classical and quantum attacks on CAST-256. "Q0" indicates a classical attack. Quantum time complexities are counted in equivalent computations of the corresponding cipher.

Setting	Type	Rounds	Data	Time	QRACM	Reference
Q0	Linear	24	$2^{124.1}$	$2^{156.52}$		[WWH08]
	Multidimensional ZC	28	$2^{98.8}$	$2^{246.90}$		[BLNW12]
	Multiple ZC	29	$2^{123.2}$	$2^{218.1}$		[WWBC14]
	Linear	32	$2^{126.8}$	$2^{251.00}$		[ZWW14]
Q1	Grover's search	24	2	$2^{129.65}$	None	
Q2	Grover-meet-Simon	23	$2^{123.7}$	$2^{123.7}$	None	[SCQ+23]
Q1	Linear + QFT + Simon	24	2^{128}	$2^{124.5}$	2^{128}	Section 7
Q1	Linear + QFT + Linear	24	2^{128}	$2^{128.68}$	2^{128}	Section 8

In this section, we show how to combine a Simon-based distinguisher and a linear key-recovery on CAST-256 to achieve a 24-round attack. This attack will be subsequently modified in the next section by using a linear distinguisher instead.

7.2 Distinguisher

We use the 17-round distinguisher of [SCQ+23], which is represented in Figure 7. This distinguisher calls Simon's algorithm on a 32-bit function which contains two calls to the cipher's inverse. More precisely, if E' is the 17-round reduced cipher, the periodic function is:

$$x \mapsto f(x, \alpha_0) \oplus f(x, \alpha_1) = (E')^{-1}(c, c, \alpha_0, x)_3 \oplus (E')^{-1}(c, c, \alpha_1, x)_3 \tag{50}$$

where c, α_0, α_1 are constants. We refer to [SCQ+23] for the proof.

Similarly to Subsection 6.5, we extend this 17-round distinguisher into a Grover-meet-Simon distinguisher, that also guesses $3 \times 37 = 111$ bits of key (in blue in Figure 7), in

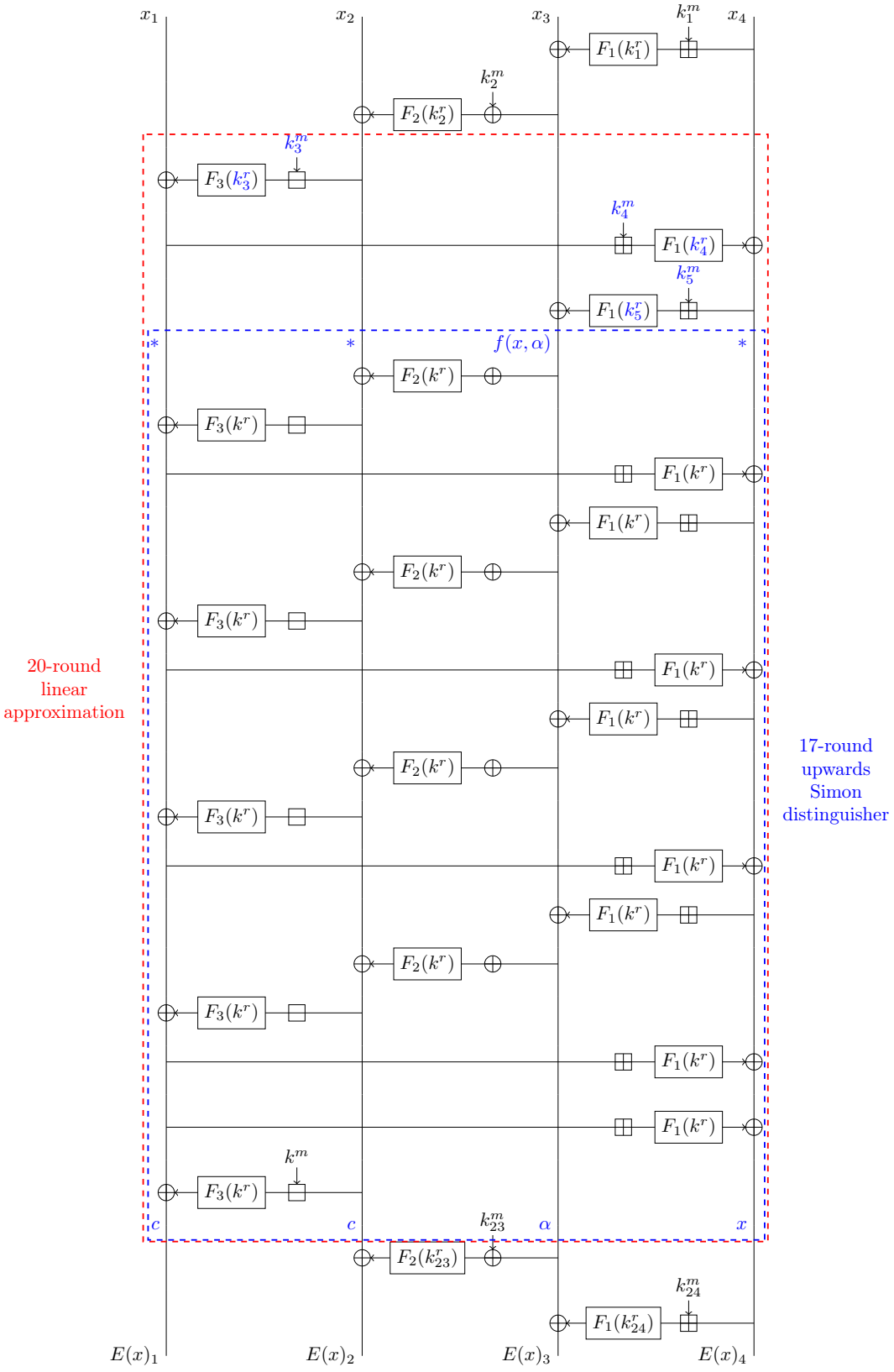

Figure 7: Path of the attack on 24-round CAST-256. The three guessed keys at the top, in order to extend the distinguisher by 3 rounds, are highlighted in blue.

order to reach 20 rounds. We use $2^8 = 256$ queries in Simon's algorithm in order to detect a periodic function with large probability. This will be our implementation of Dist for this section. We upper bound its cost by $\frac{\pi}{4}2^{111/2} \times 2^{10}Q \simeq 2^{65.2}Q$.

Linear Approximation. We use the following linear approximation for F_2 found by Wang et al. [WWH08]: $0 \to 03400000$ which holds with a bias of $2^{-12.91}$ (i.e., correlation $2^{-11.91}$). Then the linear approximation $(0, 03400000, 0, 0) \to (0, 03400000, 0, 0)$ holds for the 20 rounds of the distinguisher with correlation $(2^{-11.91})^4 = 2^{-47.64}$. For the right key guess, when using at least 2^{127} data, by Equation 17 we can lower bound the experimental correlation by $2^{-47.64} - 2 \times 2^{-127/2} \simeq 2^{-47.64}$.

7.3 FFT Key-recovery Process

We append two rounds both in input and output of the distinguisher, as shown in Figure 7. Hence we target a total of 24 rounds or 6 quad-rounds. We denote the input of this 24-round CAST as $x := (x_1, x_2, x_3, x_4)$ and the output as $E(x) := E(x)_1, E(x)_2, E(x)_3, E(x)_4$. Let $\alpha := 03400000$. The value in branch 1 after the first two rounds can be expressed as:

$$x_2 \oplus F_2(k_2^r, k_2^m \oplus x_3 \oplus F_1(k_1^r, k_1^m + x_4)) \ , \tag{51}$$

and the value in branch 1 before the last two rounds is:

$$E(x)_2 \oplus F_2(k_{23}^r, k_{23}^m \oplus E(x)_3 \oplus F_1(k_{24}^r, k_{24}^m + E(x)_4)) \ . \tag{52}$$

Considering a database of plaintext-ciphertext entries $\mathcal{D}$, we have the following expression for the experimental correlation, depending on the chosen value $(z_1^m, z_2^m, z_{23}^m, z_{24}^m)$ for the 128-bit key $k_1^m, k_2^m, k_{23}^m, k_{24}^m$ and the value $(z_1^r, z_2^r, z_{23}^r, z_{24}^r)$ for the keys $k_1^r, k_2^r, k_{23}^r, k_{24}^r$:

$$\begin{aligned}\widehat{\mathrm{cor}}(z_1^m, z_2^m, z_{23}^m, z_{24}^m, z_1^r, z_2^r, z_{23}^r, z_{24}^r) = \\ \frac{1}{|\mathcal{D}|} \sum_{\substack{(x_1,x_2,x_3,x_4,\\ E(x)_1,E(x)_2,E(x)_3,E(x)_4)\in\mathcal{D}}} (-1)^{\alpha\cdot(x_2\oplus F_2(z_2^r, z_2^m\oplus x_3\oplus F_1(z_1^r, z_1^m+x_4)))} \\ (-1)^{\alpha\cdot(E(x)_2\oplus F_2(k_{23}^r, z_{23}^m\oplus E(x)_3\oplus F_1(k_{24}^r, z_{24}^m+E(x)_4)))} \ .\end{aligned} \tag{53}$$

By indexing the database on $x_3, x_4, E(x)_3, E(x)_4$, we can define a function $\mathcal{D}(u, v, w, t)$ that, on input a 32×4-bit value representing $x_3, x_4, E(x)_3, E(x)_4$, outputs a 32-bit value representing $E(x)_2 \oplus x_2$.

The experimental correlation becomes the convolution of two functions $f_{z_1^r, z_2^r, z_{23}^r, z_{24}^r}$ and g over $(\mathbb{F}_2^{32})^2 \times (\mathbb{Z}_{2^{32}})^2$:

$$\begin{cases} g(u, v, w, t) = \mathbf{1}[(u, v, w, t) \in \mathcal{D}](-1)^{\alpha\cdot\mathcal{D}(u,v,w,t)} \\ f_{z_1^r, z_2^r, z_{23}^r, z_{24}^r}(u, v, w, t) = (-1)^{\alpha\cdot F_2(z_2^r, u\oplus F_1(z_1^r, w))}(-1)^{\alpha\cdot F_2(z_{23}^r, v\oplus F_1(z_{24}^r, t))} \end{cases} \tag{54}$$

When the full codebook is available, the amount of data available after constructing the database is $|\mathcal{D}| = 2^{128-0.66}$ following a similar analysis as in Section 6.

7.4 Computation of the Correlations

We now need to implement the unitary fFourier:

$$|z_1^r, z_2^r, z_{23}^r, z_{24}^r\rangle |u, v, w, t\rangle \mapsto |z_1^r, z_2^r, z_{23}^r, z_{24}^r\rangle |u, v, w, t\rangle \left(\frac{\widehat{f}_{z_1^r, z_2^r, z_{23}^r, z_{24}^r}(u, v, w, t)}{G} |0\rangle + |*\rangle \right) \ . \tag{55}$$

Like in Section 6, f is a product of two independent functions, so its FFT as well. Let us define:

$$f'_{z_1^r,z_2^r}(u,w) = (-1)^{\alpha\cdot F_2(z_2^r,u\oplus F_1(z_1^r,w))} \tag{56}$$

then we have:

$$\widehat{f}_{z_1^r,z_2^r,z_{23}^r,z_{24}^r}(u,v,w,t) = \widehat{f'}_{z_1^r,z_2^r}(u,w)\widehat{f'}_{z_{23}^r,z_{24}^r}(v,t) \ . \tag{57}$$

We focus on the Fourier transform of f':

$$\begin{aligned}
\widehat{f'}_{z_1^r,z_2^r}(u,w) &= \sum_{y\in\mathbb{Z}_{2^{32}}} \exp(2\iota\pi yw/2^{32}) \underbrace{\sum_{x\in\mathbb{F}_2^{32}} (-1)^{x\cdot u}(-1)^{\alpha\cdot F_2(z_2^r,x\oplus F_1(z_1^r,y))}}_{\text{Change of variable: } x\leftarrow x\oplus F_1(z_1^r,y)} \\
&= \sum_{y\in\mathbb{Z}_{2^{32}}} \exp(2\iota\pi yw/2^{32}) \sum_{x\in\mathbb{F}_2^{32}} (-1)^{F_1(z_1^r,y)\cdot u}(-1)^{x\cdot u}(-1)^{\alpha\cdot F_2(z_2^r,x)} \\
&= \underbrace{\left(\sum_{y\in\mathbb{Z}_{2^{32}}} \exp(2\iota\pi yw/2^{32})(-1)^{F_1(z_1^r,y)\cdot u}\right)}_{:=p_{z_1^r}(u,w)} \underbrace{\left(\sum_{x\in\mathbb{F}_2^{32}} (-1)^{x\cdot u}(-1)^{\alpha\cdot F_2(z_2^r,x)}\right)}_{:=q_{z_2^r}(u)} \ .
\end{aligned}$$

From Section 6, we know that $|q_{z_2^r}(u)| \le 2^{20}$ for all u. Furthermore, we compute that for all z_2^r, $|q_{z_2^r}(0)| = |\sum_x (-1)^{\alpha\cdot F_2(0,x)}| \simeq 2^{16.62}$.

Bound on p. For p we notice first that $\forall z_1^r, p_{z_1^r}(0,0) = 2^{32}$. For non-zero (u,w), we can compute a more precise bound. We start by a triangle inequality:

$$|p_{z_1^r}(u,w)| \le \left|\sum_{y\in\mathbb{Z}_{2^{32}}} \cos\left(\frac{2yw}{2^{32}}\right)(-1)^{F_1(z_1^r,y)\cdot u}\right| + \left|\sum_{y\in\mathbb{Z}_{2^{32}}} \sin\left(\frac{2yw}{2^{32}}\right)(-1)^{F_1(z_1^r,y)\cdot u}\right| \ .$$

We consider a random $(u,w) \neq (0,0)$. For fixed y, we introduce a random variable $X_y := \cos(yw/2^{32})(-1)^{F_1(z_1^r,y)\cdot u}$ which has average 0 over (u,w), and values in the interval $[-1;1]$. Likewise we introduce a random variable $Y_y := \sin(yw/2^{32})(-1)^{F_1(z_1^r,y)\cdot u}$. If all random variables X_y (resp. Y_y) were independent, using Hoeffding's inequality, we would obtain:

$$\begin{aligned}
\forall t, \Pr_{u,w}(|p_{z_1^r}(u,w)| \ge 2t) &\le \Pr_{u,w}(|\sum_y X_y| + |\sum_y Y_y| \ge 2t) \\
&= \Pr_{u,w}(|\sum_y X_y| + |\sum_y Y_y| \ge 2t, |\sum_y Y_y| < t) \\
&\quad + \Pr_{u,w}(|\sum_y X_y| + |\sum_y Y_y| \ge 2t, |\sum_y Y_y| \ge t) \\
&\le \Pr_{u,w}(|\sum_y X_y| \ge t) + \Pr_{u,w}(|\sum_y Y_y| \ge t) \\
&\le 2e^{-2t^2/(4\times 2^{32})} \ .
\end{aligned}$$

We will make the heuristic assumption that the variables behave as independent, and find a bound t so that the inequality is satisfied for *all* u, w, z_1^r (space of size 2^{64+5}) with probability $1-2^{-4}$. This puts the following constraint on t:

$$2e^{-t^2/2^{33}}2^{69} \le 2^{-4} \implies \frac{t^2}{2^{33}} \ge 74\log 2 \implies t \simeq 2^{23.34} \ .$$

Lemma 10. *Under the heuristic of independence in the sum for p, with probability* $1-2^{-4}$*:*

$$\forall u, w, z_1^r, z_2^r, |\widehat{f'}_{z_1^r, z_2^r}(x, y)| \leq \max(2^{20.34} \times 2^{20}, 2^{32} \times 2^{16.62}) = 2^{48.62} \ .$$

Consequently the bound G for $|\widehat{f}|$ can be taken as $(2^{48.62})^2 = 2^{97.2}$. This bound is actually much larger than if f behaved as a random function; the 0 Fourier coefficient of f is higher, meaning that the function is slightly unbalanced.

Computation of $\widehat{f}$. We can precompute the functions $p_{z_1^r}$ and $q_{z_2^r}$ using around $2^{10} \times 2^{96}$ classical floating-point operations, and store their values in around 2^{64} QRACM. Then we use Lemma 7 to implement the computation of $\widehat{f}$ in the amplitudes. These costs of precomputation remain negligible with respect to the rest of the attack.

Implementation of CorComp. By Lemma 6, we can compute the correlation state with gate count:

$$\left(\frac{\pi}{2}\frac{G}{2^{n/2}} + 3\right)(10Q) = 2^{38.2}Q \ , \tag{58}$$

where we have counted 4 QRACM queries for the computation of $\widehat{f}$.

7.5 Cost of the Attack

Similarly to Subsection 6.5, we can directly use Theorem 3 with:

$$k^{\text{in}} = k_1^r, k_2^r, k_{23}^r, k_{24}^r, \quad k^{\text{out}} = k_1^m, k_2^m, k_{23}^m, k_{24}^m, \quad |k^{\text{in}}| = 20, \quad |k^{\text{out}}| = 128 \ . \tag{59}$$

Using Theorem 3, the entire key-recovery attack performs:

$$2\left\lceil 1.21 \times 2^{47.64} \times 2^{10} \times 2^{0.33} \right\rceil = 2^{59.3} \tag{60}$$

calls to both CorComp and Dist in order to retrieve the key with probability at least 0.5. The gate count is given by:

$$2^{59.3} \times \left(2^{38.2}Q + 2^{65.2}Q\right) = 2^{124.5}Q \ . \tag{61}$$

Despite the large value of the bound G that we had to use, the computation of Corcomp is still inconsequential, and the complexity is dominated by the distinguisher. Here, our advantage with respect to the Grover-meet-Simon attack, which comes from the large correlation (reducing the number of outer iterates), allows us to increase the number of attacked rounds.

8 Quantum Attack on CAST-256 with LinDist

For comparison, this section shows an attack on the 24-round CAST-256 when a linear distinguisher (LinDist of Lemma 8) is used instead of the Simon-based distinguisher. The resulting attack is slower than the attack in the previous section, but still faster than the exhaustive key search with Grover's algorithm.

Comparison with the Previous Section. The linear approximation we use here is the same as before, i.e., the 20-round approximation of the absolute correlation $2^{-47.64}$, which applies for the cipher shown in Figure 7 except for the first two rounds and the last two rounds. The distinguisher LinDist is also applied to this 20-round part.

Cost of the Distinguisher. We apply LinDist with $\ell = 2^{6.08}n = 2^{13.08}$. The ELP is approximately lower bounded by $2^{-47.64}$. By Lemma 8, the gate count of LinDist is approximately upper bounded by $2^3 \cdot \left\lceil 16\pi/\sqrt{ELP} \right\rceil \cdot 2^{12} \cdot Q \leq 2^{69.38}Q$.

Cost of Corcomp. Since the cost of Corcomp does not depend on distinguishers, it is the same as before, i.e., $2^{38.2}Q$.

Overall Complexity. Before running quantum algorithms, the full codebook is queried and stored in QRAM, together with the data structure $\mathcal{D}$ of size $\simeq 2^{128-0.66}$. As before, by Theorem 3, the attack calls LinDist and Corcomp at most $2\left\lceil 1.21 \times 2^{10} \times 2^{47.64} \times 2^{0.33} \right\rceil \leq 2^{59.3}$ times. Hence the overall gate count is at most

$$2^{59.3}\left(2^{69.38}Q + 2^{38.2}Q\right) \simeq 2^{128.68}Q.$$

This is worse than the attack in the previous section by a few bits. Still, it is slightly faster than the Grover search, which requires at least $2 \times 2 \times 2^{128} = 2^{130}$ encryptions, considering the cost of uncomputation and that the key length is twice as large as the block lengths.

Remark 9. Note that the precision of LinDist to approximate Dist is sufficiently high: Since LinDist is called in the attack at most $2^{59.3}$ times, the parameter ℓ we use in the attack ($\ell = 2^{13.08}$) is larger than the right hand side of Equation 35.

9 Conclusion

In this paper, we showed how to combine a linear key-recovery attack based on the quantum Fourier transform with a distinguisher (related or not), in order to speed up some quantum key-recovery attacks.

Although our concrete results remain relatively modest, such as a speedup of a factor 2^3 of a 7-round attack on CAST-128, and an improvement by 1 round of a quantum attack on CAST-256, we believe that this framework may lead to further applications, by leveraging *multiple* cryptanalytic properties at the same time, which seems to be a first in quantum cryptanalysis. More generally, as there is no generic limitation to quadratic speedups with this approach, we believe it is very promising.

A problem that remains unsolved (already noticed in [Sch23]) is how to use *zero-correlation* linear hulls. Indeed, what the QFT-based method gives in such a case is a quantum state where the good key does *not* appear, rather than a state in which it appears with good probability. In our case we could use the zero-correlation property as a distinguisher, but we wouldn't gain anything more from it.

Acknowledgements.

This work was initiated during the Dagstuhl Seminar 23421 "Quantum cryptanalysis". The authors would like to thank Gorjan Alagic, Paul Frixons, Christian Majenz, María Naya-Plasencia and Yu Sasaki for preliminary discussions and comments.

This work has been partially supported by the French Agence Nationale de la Recherche through the QATS project under Contract ANR-24-CE39-7894-01, and through the France 2030 program under grant agreement No. ANR-22-PETQ-0007 EPiQ and No. ANR-22-PETQ-0008 PQ-TLS.

References

[ABKM22] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the Even-Mansour cipher. In *EUROCRYPT (3)*, volume 13277 of *Lecture Notes in Computer Science*, pages 458–487. Springer, 2022.

[Ada97a] Carlisle M. Adams. Constructing symmetric ciphers using the CAST design procedure. *Des. Codes Cryptogr.*, 12(3):283–316, 1997.

[Ada97b] C.M. Adams. The CAST-128 encryption algorithm. RFC 2144 https://datatracker.ietf.org/doc/html/rfc2144, May 1997.

[AG99] Dr. Carlisle Adams and Jeff Gilchrist. The CAST-256 Encryption Algorithm. RFC 2612, June 1999.

[BBC+21] Ritam Bhaumik, Xavier Bonnetain, André Chailloux, Gaëtan Leurent, María Naya-Plasencia, André Schrottenloher, and Yannick Seurin. QCB: efficient quantum-secure authenticated encryption. In *ASIACRYPT (1)*, volume 13090 of *Lecture Notes in Computer Science*, pages 668–698. Springer, 2021.

[BHMT02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[BHN+19] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In *ASIACRYPT (1)*, volume 11921 of *Lecture Notes in Computer Science*, pages 552–583. Springer, 2019.

[BJ22] Xavier Bonnetain and Samuel Jaques. Quantum period finding against symmetric primitives in practice. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):1–27, 2022.

[BKPS91] Lawrence Brown, Matthew Kwan, Josef Pieprzyk, and Jennifer Seberry. Improving resistance to differential cryptanalysis and the redesign of LOKI. In *ASIACRYPT*, volume 739 of *Lecture Notes in Computer Science*, pages 36–50. Springer, 1991.

[BLNW12] Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. Integral and multidimensional linear distinguishers with correlation zero. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 244–261. Springer, 2012.

[BN16] Céline Blondeau and Kaisa Nyberg. Improved parameter estimates for correlation and capacity deviates in linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2016(2):162–191, 2016.

[BN17] Céline Blondeau and Kaisa Nyberg. 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.

[BNS19a] Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. On quantum slide attacks. In *SAC*, volume 11959 of *Lecture Notes in Computer Science*, pages 492–519. Springer, 2019.

[BNS19b] Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. Quantum security analysis of AES. *IACR Trans. Symmetric Cryptol.*, 2019(2):55–93, 2019.

[Bon21] Xavier Bonnetain. Tight bounds for Simon's algorithm. In *LATINCRYPT*, volume 12912 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2021.

[BSS22] Xavier Bonnetain, André Schrottenloher, and Ferdinand Sibleyras. Beyond quadratic speedups in quantum attacks on symmetric schemes. In *EUROCRYPT (3)*, volume 13277 of *Lecture Notes in Computer Science*, pages 315–344. Springer, 2022.

[CHLS20] Carlos Cid, Akinori Hosoyamada, Yunwen Liu, and Siang Meng Sim. Quantum cryptanalysis on contracting Feistel structures and observation on related-key settings. In *INDOCRYPT*, volume 12578 of *Lecture Notes in Computer Science*, pages 373–394. Springer, 2020.

[CLS22] Federico Canale, Gregor Leander, and Lukas Stennes. Simon's algorithm and symmetric crypto: Generalizations and automatized applications. In *CRYPTO (3)*, volume 13509 of *Lecture Notes in Computer Science*, pages 779–808. Springer, 2022.

[CSQ07] Baudoin Collard, François-Xavier Standaert, and Jean-Jacques Quisquater. Improving the time complexity of matsui's linear cryptanalysis. In *ICISC*, volume 4817 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2007.

[DLW19] Xiaoyang Dong, Zheng Li, and Xiaoyun Wang. Quantum cryptanalysis on some generalized Feistel schemes. *Sci. China Inf. Sci.*, 62(2):22501:1–22501:12, 2019.

[FN20] Antonio Flórez-Gutiérrez and María Naya-Plasencia. Improving key-recovery in linear attacks: Application to 28-round PRESENT. In *EUROCRYPT (1)*, volume 12105 of *Lecture Notes in Computer Science*, pages 221–249. Springer, 2020.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219. ACM, 1996.

[HK20] Samir Hodzic and Lars R. Knudsen. A quantum distinguisher for 7/8-round SMS4 block cipher. *Quantum Inf. Process.*, 19(11):411, 2020.

[Hos23] Akinori Hosoyamada. Quantum speed-up for multidimensional (zero correlation) linear distinguishers. In *ASIACRYPT (3)*, volume 14440 of *Lecture Notes in Computer Science*, pages 311–345. Springer, 2023.

[Hos24] Akinori Hosoyamada. Quantum algorithms for fast correlation attacks on LFSR-based stream ciphers, 2024.

[HS18] Akinori Hosoyamada and Yu Sasaki. Quantum demiric-selçuk meet-in-the-middle attacks: Applications to 6-round generic Feistel constructions. In *SCN*, volume 11035 of *Lecture Notes in Computer Science*, pages 386–403. Springer, 2018.

[HS20] Akinori Hosoyamada and Yu Sasaki. Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In *EUROCRYPT (2)*, volume 12106 of *Lecture Notes in Computer Science*, pages 249–279. Springer, 2020.

[IHM+19] Gembu Ito, Akinori Hosoyamada, Ryutaroh Matsumoto, Yu Sasaki, and Tetsu Iwata. Quantum chosen-ciphertext attacks against Feistel ciphers. In *CT-RSA*, volume 11405 of *Lecture Notes in Computer Science*, pages 391–411. Springer, 2019.

[IS12] Takanori Isobe and Kyoji Shibutani. All subkeys recovery attack on block ciphers: Extending meet-in-the-middle approach. In *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 202–221. Springer, 2012.

[IS13] Takanori Isobe and Kyoji Shibutani. Generic key recovery attack on Feistel scheme. In *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 464–485. Springer, 2013.

[KLLN16a] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[KLLN16b] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Quantum differential and linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2016(1):71–94, 2016.

[KM10] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *ISIT*, pages 2682–2685. IEEE, 2010.

[KM12] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *ISITA*, pages 312–316. IEEE, 2012.

[LM17] Gregor Leander and Alexander May. Grover meets Simon - quantumly attacking the FX-construction. In *ASIACRYPT (2)*, volume 10625 of *Lecture Notes in Computer Science*, pages 161–178. Springer, 2017.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[Mat94] Mitsuru Matsui. The first experimental cryptanalysis of the data encryption standard. In *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 1994.

[MS22] Alexander May and Lars Schlieper. Quantum period finding is compression robust. *IACR Trans. Symmetric Cryptol.*, 2022(1):183–211, 2022.

[NC02] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[NIDI19] Boyu Ni, Gembu Ito, Xiaoyang Dong, and Tetsu Iwata. Quantum attacks against type-1 generalized Feistel ciphers and applications to CAST-256. In *INDOCRYPT*, volume 11898 of *Lecture Notes in Computer Science*, pages 433–455. Springer, 2019.

[ÖBR23] Betül Askin Özdemir, Tim Beyne, and Vincent Rijmen. Multidimensional linear cryptanalysis of Feistel ciphers. *IACR Trans. Symmetric Cryptol.*, 2023(4):1–27, 2023.

[Sch23] André Schrottenloher. Quantum linear key-recovery attacks using the QFT. In *CRYPTO (5)*, volume 14085 of *Lecture Notes in Computer Science*, pages 258–291. Springer, 2023.

[SCQ+23] Hong-Wei Sun, Bin-Bin Cai, Su-Juan Qin, Qiao-Yan Wen, and Fei Gao. Quantum attacks on type-1 generalized Feistel schemes. *Advanced Quantum Technologies*, 6(10):2300155, 2023.

[SF97] Kouichi Sakurai and Souichi Furuya. Improving linear cryptanalysis of LOKI91 by probabilistic counting method. In *FSE*, volume 1267 of *Lecture Notes in Computer Science*, pages 114–133. Springer, 1997.

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.

[SLSB19] Yuval R Sanders, Guang Hao Low, Artur Scherer, and Dominic W Berry. Black-box quantum state preparation without arithmetic. *Physical review letters*, 122(2):020502, 2019.

[SS24] André Schrottenloher and Marc Stevens. Quantum procedures for nested search problems: with applications in cryptanalysis. *IACR Commun. Cryptol.*, 1(3):9, 2024.

[TSM94] Toshio Tokita, Tohru Sorimachi, and Mitsuru Matsui. Linear cryptanalysis of LOKI and s^2DES. In *ASIACRYPT*, volume 917 of *Lecture Notes in Computer Science*, pages 293–303. Springer, 1994.

[WWBC14] Long Wen, Meiqin Wang, Andrey Bogdanov, and Huaifeng Chen. General application of FFT in cryptanalysis and improved attack on CAST-256. In *INDOCRYPT*, volume 8885 of *Lecture Notes in Computer Science*, pages 161–176. Springer, 2014.

[WWH08] Meiqin Wang, Xiaoyun Wang, and Changhui Hu. New linear cryptanalytic results of reduced-round of CAST-128 and CAST-256. In *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 429–441. Springer, 2008.

[XWY+24] Zejun Xiang, Xiaoyu Wang, Bo Yu, Bing Sun, Shasha Zhang, Xiangyong Zeng, Xuan Shen, and Nian Li. Links between quantum distinguishers based on Simon's algorithm and truncated differentials. *IACR Trans. Symmetric Cryptol.*, 2024(2):296–321, 2024.

[ZWW14] Jingyuan Zhao, Meiqin Wang, and Long Wen. Improved linear cryptanalysis of CAST-256. *J. Comput. Sci. Technol.*, 29(6):1134–1139, 2014.

A Proof of Lemma 8

To show Lemma 8, we need two lemmas below.

Lemma 11. *Let* $\kappa := |k^{\text{in}}| + |k^{\text{out}}|$. *If* $n \geq 3$, $T = \frac{3}{8}\sqrt{ELP}$, *and* $1 \gg \sqrt{ELP} \geq 2^{-n/2} \cdot (4\sqrt{2\kappa})$ *holds, then*

$$3\sqrt{ELP} \geq |\mathrm{cor}(k^{\text{in}}, k^{\text{out}})| \geq \sqrt{ELP}/2 \quad (\Leftrightarrow 8T \geq |\mathrm{cor}(k^{\text{in}}, k^{\text{out}})| \geq (4/3)T)$$

and

$$|\mathrm{cor}(z^{\text{in}}, z^{\text{out}})| \leq \sqrt{ELP}/4 = (2/3)T$$

hold for all $(z^{\text{in}}, z^{\text{out}}) \neq (k^{\text{in}}, k^{\text{out}})$ *with probability at least around* $0.95 - (2/e)^{\kappa}\sqrt{2\kappa}$ *over the choice of the key.*

Proof. By the assumption $\sqrt{ELP} \geq 2^{-n/2} \cdot (4\sqrt{2\kappa})$ (and $\kappa \geq |k^{\mathrm{out}}| = 1$), we have

$$\sqrt{ELP} > 4 \cdot 2^{-n/2} = 4/\sqrt{N}.$$

This inequality and Equation 17 imply

$$\begin{aligned}
&\Pr_K\left[|\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})| < \sqrt{ELP}/2 \text{ or } |\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})| > 3\sqrt{ELP}\right] \\
&\quad \leq \Pr_K\left[|\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})| < \sqrt{ELP} - 2/\sqrt{N} \text{ or } |\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})| > \sqrt{ELP} + 2/\sqrt{N}\right] \\
&\quad \leq 0.05. \qquad (62)
\end{aligned}$$

In addition, by the wrong key hypothesis, the value $\mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})$ follows $\mathcal{N}(0, 2^{-n})$ for $(z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}})$. Hence, it holds that

$$\begin{aligned}
&\Pr_K\left[\exists (z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}}), |\mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})| > \sqrt{ELP}/4\right] \\
&\quad \leq 2 \Pr_K\left[\exists (z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}}), \mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}}) > 2^{-n/2}\sqrt{2\kappa}\right] \\
&\quad \leq 2 \sum_{(z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}})} \Pr_K\left[\mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}}) > 2^{-n/2}\sqrt{2\kappa}\right] \\
&\quad \simeq \frac{2^{\kappa+1}}{\sqrt{2\pi}} \int_{\sqrt{2\kappa}}^{\infty} e^{-x^2/2} dx \leq 2^{\kappa} \int_{\sqrt{2\kappa}}^{\infty} \frac{x}{\sqrt{2\kappa}} e^{-x^2/2} dx = -\frac{2^{\kappa}}{\sqrt{2\kappa}} \int_{\sqrt{2\kappa}}^{\infty} \left(e^{-x^2/2}\right)' dx \\
&\quad = \frac{2^{\kappa}}{\sqrt{2\kappa}} e^{-\kappa}. \qquad (63)
\end{aligned}$$

The claim follows from Equation 62 and Equation 63. □

Lemma 12. *Suppose* $T = \frac{3}{8}\sqrt{ELP}$ *and* $q = \left\lceil 16\pi/\sqrt{ELP} \right\rceil$. *In addition, assume* $ELP \ll 1$, $8T \geq |\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})| \geq (4/3)T$, *and* $|\mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})| \leq (2/3)T$ *for all* $(z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}})$. *If* QC_q *runs on* $f_{z^{\mathrm{in}}, z^{\mathrm{out}}}$ *under this assumption, then with probability of at least* 0.8 *it outputs an integer* $\tilde{X}$ *such that* $|2\tilde{X}/2^n - 1| \geq T$ *if* $(z^{\mathrm{in}}, z^{\mathrm{out}}) = (k^{\mathrm{in}}, k^{\mathrm{out}})$ *and* $|2\tilde{X}/2^n - 1| < T$ *if* $(z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}})$.

Proof. Let $X := \#\{x : f_{z^{\mathrm{in}}, z^{\mathrm{out}}}(x) = 1\} (= \#\{x : \alpha \cdot F^L_{z^{\mathrm{in}}}(x \oplus z^{\mathrm{out}}) = \beta \cdot F^R_{z^{\mathrm{in}}}(E_K(x))\})$. Then we have

$$X = \frac{1 + \mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})}{2} \cdot 2^n \text{ and } 2^n - X = \frac{1 - \mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})}{2} \cdot 2^n.$$

As we are assuming $8T \geq |\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})|$ holds, $|\mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})| \leq (2/3)T$ for all $(z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}})$, and that $\sqrt{ELP} \ll 1$,

$$X(2^n - X) = \frac{1 - \mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})^2}{4} \cdot 2^{2n} \simeq 2^{2n-2}$$

holds.

By the explanation in Subsection 2.6 and the assumption $\sqrt{ELP} \ll 1$, we have

$$\begin{aligned}
\left|\tilde{X} - X\right| &\leq 2\pi\sqrt{X(2^n - X)/q^2} + (\pi \cdot 2^{n/2}/q)^2 \\
&\lessapprox 2\pi 2^{n-1} / \left(16\pi/\sqrt{ELP}\right) + \left(\pi \cdot 2^{n/2}/(16\pi/\sqrt{ELP})\right)^2 \\
&= 2^n\sqrt{ELP}/16 + (2^n\sqrt{ELP}/256) \cdot \sqrt{ELP} \\
&\simeq 2^n\sqrt{ELP}/16. \qquad (64)
\end{aligned}$$

Multiplying both sides of the inequality $2^n\sqrt{ELP}/16 > \left|\tilde{X} - X\right|$ by $2/2^n$, we obtain

$$\begin{aligned}\sqrt{ELP}/8 > (2/2^n)\left|\tilde{X} - X\right| &= \left|(2\tilde{X}/2^n - 1) - (2X/2^n - 1)\right| \\ &= \left|(2\tilde{X}/2^n - 1) - \mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})\right|,\end{aligned}$$

which implies

$$\left|(2\tilde{X}/2^n - 1) - \mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})\right| < T/3. \tag{65}$$

Due to the current assumption that $|\mathrm{cor}(k^{\mathrm{in}}, k^{\mathrm{out}})| \geq (4/3)T$ and $|\mathrm{cor}(z^{\mathrm{in}}, z^{\mathrm{out}})| \leq (2/3)T$ for all $(z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}})$, the claim of the lemma follows. □

Proof of Lemma 8. The gate count to implement f is at most about $4Q$, so the claim about the gate count immediatley follows.

Let U_1 and U_2 be the uniary operators corresponing to Step 1 and Step 2 of $\mathsf{LinDist}$, respectively. As Step 3 is the uncomputaion of Step 1, we have

$$\mathsf{LinDist} = U_1^\dagger U_2 U_1.$$

In adition, let U_{flip} be the unitary operator such that

$$U_{\mathsf{flip}} |b\rangle = |b \oplus 1\rangle$$

for $b \in \{0, 1\}$.

We first analyze the behavior of $\mathsf{LinDist}$ on the right key. Suppose that the assumption of Lemma 12 holds and that we run $\mathsf{LinDist}$ on $(z^{\mathrm{in}}, z^{\mathrm{out}}) = (k^{\mathrm{in}}, k^{\mathrm{out}})$, measuring the state just after Step 1 (i.e., we start from the non-superposed state $|k^{\mathrm{in}}, k^{\mathrm{out}}, b\rangle$ for some $b \in \{0, 1\}$ and measure $U_1 |k^{\mathrm{in}}, k^{\mathrm{out}}, b\rangle$). Let cnt be the number of i satisfying $|2\tilde{X}_i/2^n - 1| \geq T$. In addition, let Π_{good} (resp., Π_{bad}) be the projector onto the space of quantum states corresponding to $\mathsf{cnt} \geq \ell/2$ (resp., $\mathsf{cnt} < \ell/2$). Then, cnt follows the binomial distribution $B(\ell, \mathsf{p})$ for some $\mathsf{p} \geq 0.8$ by Lemma 12, and we have

$$\begin{aligned}&\|\Pi_{\mathsf{bad}} U_1 |k^{\mathrm{in}}, k^{\mathrm{out}}, b\rangle\|^2 \\ &\quad \Pr\left[\text{Step 2 fails (i.e., judges that } (z^{\mathrm{in}}, z^{\mathrm{out}}) \neq (k^{\mathrm{in}}, k^{\mathrm{out}}))\right] \\ &\quad = \Pr\left[\mathsf{cnt} < \ell/2\right] \leq \Pr\left[\mathsf{cnt} < \frac{5}{8}\mathsf{p}\ell\right] \underset{(*)}{\leq} \left(\frac{e^{-3/8}}{(5/8)^{5/8}}\right)^{\mathsf{p}\ell} \leq \left(\frac{1}{2}\right)^{0.09\ell},\end{aligned} \tag{66}$$

where $(*)$ follows from the Chernoff bound. This implies

$$\begin{aligned}
&\left\| \mathsf{LinDist}\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle - \underbrace{\mathsf{Dist}\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle}_{=|k^{\mathrm{in}},k^{\mathrm{out}},b\oplus 1\rangle = U_{\mathsf{flip}}|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle} \right\| \\
&= \left\| U_1^\dagger U_2(\Pi_{\mathsf{good}}+\Pi_{\mathsf{bad}})U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle - \underbrace{U_{\mathsf{flip}}}_{\text{commutative with } U_1 \text{ and } U_1^\dagger}|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| \\
&\leq \left\| U_1^\dagger \underbrace{U_2\Pi_{\mathsf{good}}}_{=U_{\mathsf{flip}}\Pi_{\mathsf{good}}} U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle - U_1^\dagger U_{\mathsf{flip}} U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| + \left\| U_1^\dagger U_2\Pi_{\mathsf{bad}}U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| \\
&= \left\| U_1^\dagger U_{\mathsf{flip}}\Pi_{\mathsf{good}}U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle - U_1^\dagger U_{\mathsf{flip}} \underbrace{U_1}_{=(\Pi_{\mathsf{good}}+\Pi_{\mathsf{bad}})U_1}|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| + \left\| \Pi_{\mathsf{bad}}U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| \\
&= \left\| U_1^\dagger U_{\mathsf{flip}}\Pi_{\mathsf{bad}}U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| + \left\| \Pi_{\mathsf{bad}}U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| \\
&\leq 2\left\| \Pi_{\mathsf{bad}}U_1\,|k^{\mathrm{in}},k^{\mathrm{out}},b\rangle \right\| \leq 2\left(\frac{1}{2}\right)^{0.045\ell} \qquad (67)
\end{aligned}$$

for any $b\in\{0,1\}$. We can show

$$\left\| \mathsf{LinDist}\,|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle - \mathsf{Dist}\,|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle \right\| \leq 2\left(\frac{1}{2}\right)^{0.045\ell} \qquad (68)$$

for $(z^{\mathrm{in}},z^{\mathrm{out}})\neq(k^{\mathrm{in}},k^{\mathrm{out}})$ and $b\in\{0,1\}$ in the same way. Therefore, for arbitrary quantum state of the form $|\psi\rangle=\sum_{z^{\mathrm{in}},z^{\mathrm{out}},b}\alpha_{z^{\mathrm{in}},z^{\mathrm{out}},b}\,|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle$,

$$\begin{aligned}
\|\mathsf{LinDist}\,|\psi\rangle - \mathsf{Dist}\,|\psi\rangle\| &\leq \sum_{z^{\mathrm{in}},z^{\mathrm{out}},b} |\alpha_{z^{\mathrm{in}},z^{\mathrm{out}},b}| \left\| \mathsf{LinDist}\,|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle - \mathsf{Dist}\,|z^{\mathrm{in}},z^{\mathrm{out}},b\rangle \right\| \\
&\leq \sum_{z^{\mathrm{in}},z^{\mathrm{out}},b} |\alpha_{z^{\mathrm{in}},z^{\mathrm{out}},b}| 2\left(\frac{1}{2}\right)^{0.045\ell} \\
&\leq 2^{(|k^{\mathrm{in}}|+|k^{\mathrm{out}}|)/2-0.045\ell+2}\sqrt{\sum_{z^{\mathrm{in}},z^{\mathrm{out}},b}|\alpha_{z^{\mathrm{in}},z^{\mathrm{out}},b}|^2} \\
&\leq 2^{(|k^{\mathrm{in}}|+|k^{\mathrm{out}}|)/2-0.045\ell+2}, \qquad (69)
\end{aligned}$$

where the second last inequality follows from Jensen's inequality.

Thus, the claim of Lemma 8 follows from Lemma 11. □

B Lemma 10 from [Sch23]

For completeness, we provide the following lemma, used in Subsection 5.2.

Lemma 13 ([Sch23, 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:*

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